FORM PTO-1390 (Modified) (REV 11-98) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371

ATTORNEY'S DOCKET NUMBER	
033808/027 8780	
U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR)	
09/890929	
INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE / PRIORITY DATE CLAIMED
PCT/JP00/08133	17 November 2000 / 14 December 1999

TITLE OF INVENTION
METHOD AND SYSTEM FOR DISPLAYING DENDROGRAM

APPLICANT(S) FOR DO/EO/US
HITACHI SOFTWARE ENGINEERING CO., LTD., et al.

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☐ This is an express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(1).
4. ☐ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. ☒ A copy of the International Application as filed (35 U.S.C. 371 (c) (2))
   a. ☐ is transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☒ has been transmitted by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
7. ☐ A copy of the International Search Report (PCT/ISA/210).
8. ☒ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371 (c)(3))
   a. ☐ are transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ have been transmitted by the International Bureau.
   c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
   d. ☒ have not been made and will not be made.
9. ☐ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
10. ☒ An oath or declaration of the inventor(s) (35 U.S.C. 371 (c)(4)).
11. ☐ A copy of the International Preliminary Examination Report (PCT/IPEA/409).
12. ☐ A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371 (c)(5)).

Items 13 to 20 below concern document(s) or information included:

13. ☐ An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
14. ☒ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
15. ☐ A **FIRST** preliminary amendment.
16. ☐ A **SECOND** or **SUBSEQUENT** preliminary amendment.
17. ☐ A substitute specification.
18. ☐ A change of power of attorney and/or address letter.
19. ☒ Certificate of Mailing by Express Mail
20. ☐ Other items or information:

PCT/IB/308;
PCT Request (PH1028-PCT);
Int'l Publication (WO 01/45026 A2); and
Reply Postcard

U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR	INTERNATIONAL APPLICATION NO.	ATTORNEY'S DOCKET NUMBER
09/890929	PCT/JP00/08133	033808/027 8780

21. The following fees are submitted:.

BASIC NATIONAL FEE (37 CFR 1.492 (a) (1) - (5)) :

		CALCULATIONS PTO USE ONLY
☐	Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2) paid to USPTO and International Search Report not prepared by the EPO or JPO **$1,000.00**	
☒	International preliminary examination fee (37 CFR 1.482) not paid to USPTO but Internation Search Report prepared by the EPO or JPO **$860.00**	
☐	International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO **$710.00**	
☐	International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provisions of PCT Article 33(1)-(4) **$690.00**	
☐	International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(1)-(4) **$100.00**	
	ENTER APPROPRIATE BASIC FEE AMOUNT =	**$860.00**
	Surcharge of **$130.00** for furnishing the oath or declaration later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492 (e)).	**$0.00**

CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE	
Total claims	13 - 20 =	0	x $18.00	**$0.00**
Independent claims	5 - 3 =	2	x $80.00	**$160.00**
Multiple Dependent Claims **(check if applicable).**			☒	**$270.00**
	TOTAL OF ABOVE CALCULATIONS =			**$1,290.00**
	Reduction of 1/2 for filing by small entity, if applicable. Verified Small Entity Statement must also be filed (Note 37 CFR 1.9, 1.27, 1.28) **(check if applicable).** ☐			**$0.00**
	SUBTOTAL =			**$1,290.00**
	Processing fee of **$130.00** for furnishing the English translation later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492 (f)). +			**$0.00**
	TOTAL NATIONAL FEE =			**$1,290.00**
	Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31) **(check if applicable).** ☐			**$0.00**
	TOTAL FEES ENCLOSED =			**$1,290.00**
				Amount to be: refunded $
				charged $

☐ A check in the amount of to cover the above fees is enclosed.

☒ Please charge my Deposit Account No. **03-3975** in the amount of **$1,290.00** to cover the above fees.
A duplicate copy of this sheet is enclosed.

☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **03-3975** A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

PILLSBURY WINTHROP LLP
50 Fremont Street
San Francisco, CA 94105

SIGNATURE

Robert M. Bedgood
NAME

43,488
REGISTRATION NUMBER

August 7, 2001
DATE

METHOD AND SYSTEM FOR DISPLAYING DENDROGRAM

FIELD OF THE INVENTION

The present invention relates to a method and a system for displaying data (gene expression data) obtained by hybridization with a specific biopolymer such as a gene, in a visually comprehensible format so that functions and roles of the biopolymer (gene) can readily be studied.

BACKGROUND OF THE INVENTION

With the increase in the number of species that have been determined of their genome sequences, so called genome comparison has extensively been performed. Genome comparison aims at finding facts based on gene differences among species, for example, finding genes involved in evolution, finding a collection of genes which are considered to be common to all species, or, conversely, studying the nature unique to specific species. The recent development of infrastructures such as DNA chips and DNA microarrays has changed the interest in the art of molecular biology from information of interspecies to information of intraspecies, namely coexpression analysis, and broadened the study covering from extraction of information to correlation of information, including the conventional comparison between species.

For example, if an unknown gene has an expression pattern identical to that of a known gene, the unknown gene

can be assumed to have a similar function to that of the known gene. Functional meanings of such genes and proteins are studied as function units or function groups. The interactions between the function units or function groups are also analyzed by correlating with known enzymatic reaction data or metabolism data, or more directly, by knocking out or overreacting a specific gene to eliminate or accelerate expression of the gene in order to study the direct and indirect influences on gene expression patterns of a whole collection of genes.

One successful case in this field would be the expression analysis of yeast by the group of P. Brown et al. from the Stanford University (Michel B. Eisen et al., Clustering analysis and display of genome-wide expression patterns, *Proc. Natl. Acad. Sci.* (1998), Dec 8; 95(25): 14863-8). They conducted hybridization of genes extracted from a cell in a time series using a DNA microarray, and numerated the expression levels thereof (i.e., numerated the brightness of the hybridized fluorescent signals). Based on the numerated values, genes having similar expression patterns in their gene cycles (genes having closer expression levels at some point) are clustered together.

Figure 1 is a diagram showing an exemplary display for showing similarity between expression patterns of genes according to the above-mentioned system. Information of each of the observed genes is listed on the right hand side, and a

dendrogram formed based on the expression patterns of these genes is drawn on the left hand side. The dendrogram is drawn by stepwisely joining every two most similar clusters together. The length of each branch corresponds to the distance (dissimilarity) between the two joined clusters. This displaying method allows a supposition that genes belonging to the same cluster may possibly share common functional characteristics.

In an actual analysis of gene expression patterns, enormous amount of data will be subjected to clustering. A DNA chip or DNA microarray is usually capable of detecting thousands to ten-thousands of genes at the same time. Generally, an expression of one gene may induce or inhibit an expression of another gene, forming a complicated network among genes. Therefore, if the numbers of genes to be observed are larger, more complicated and detailed gene network can be studied.

However, as the number of genes is increased, it becomes very difficult to find the functions of the entire genes. Since a dendrogram will represent several thousands to ten-thousands of genes, it is difficult from the display to judge what kind of grouping has been made. Furthermore, the lengths of branches in the resulting dendrogram generally differ depending on the type of clustering method employed. For example, when a furthest neighbor method is employed as a cluster combining algorithm, the average length of the

branches will be longer than the average length of branches resulting from a nearest neighbor method. Therefore, looking at overall dendrograms in Figure 2, a length from a root to leaves also varies depending on the clustering method. For clustering gene expression data, it is more important to find out the groupings than to observe the lengths of the branches. Accordingly, as shown in Figure 3, a dendrogram is generally displayed while a length from the root to the leaves of the dendrogram is fixed in advance. As a result, lengths of the branches are determined relative to the length of the whole dendrogram and a scale of the lengths of the branches differs depending on the clustering method.

According to the above-described method for displaying a dendrogram, when the dendrogram contains numbers of genes having similar expression patterns, the lengths of the branches will be short. When the lengths of these branches are too short relative to the length of the dendrogram, it becomes very difficult to find detailed relationship between the branches of genes as can be appreciated from a range 401 in Figure 4. According to a conventional clustering for a gene expression analysis, an interactive operation such as selecting a subtree and then subjecting the selected subtree to another clustering method, was impossible. Moreover, according to a conventional clustering for a gene expression analysis, whether the grouping was successful or not is confirmed by focusing on the functions of genes or keywords

derived from gene names to see whether relative genes are assembled in a subtree. However, when the number of genes to be analyzed is numerous, it is difficult to determine which function or keyword should be focused on.

The present invention aims at solving such conventional problems, and has an objective to provide a method and a system for displaying a dendrogram such that the state of branches of the whole dendrogram can globally be understood, and such that a detailed state of each subtree can be studied.

SUMMARY OF THE INVENTION

In order to achieve the above-mentioned objective, the present invention proposes a system for displaying a dendrogram which is provided with functions for selecting a branch in a dendrogram, displaying a subtree extending from the selected branch to the downstream leaves on a separate display window, replacing the subtree with an icon, restoring the icon to the subtree, and collecting and displaying keywords contained in the subtree. According to the present invention, subtrees of a produced dendrogram can be subjected to different clustering methods interactively. Keywords contained in the subtrees can be displayed in order to confirm success of clustering as well as to aid focusing of groupings and to aid selection of a clustering method.

Hereinafter, exemplary dendrogram displays according to a dendrogram displaying system of the invention will be

described. Herein, for clearer understanding, the invention is applied to a case of genes, although the application of the present invention is not limited to genes. The present invention can equally be applied to other biopolymers such as cDNAs, RNAs, DNA fragments or the like.

Figure 5 is a view showing an exemplary display of a dendrogram resulting from a dendrogram displaying system of the invention. The display includes a grouping algorithm selection menu 501 and a (dis)similarity selection menu 502. A dendrogram is produced by reading out gene expression data, and selecting a grouping algorithm and a type of (dis)similarity. The present system may also be capable of displaying gene information next to the leaves of the dendrogram as shown in Figure 1.

By selecting a branch in the produced dendrogram, a subtree extending from the selected branch to the downstream leaves can be made the subject of operations. Specifically, the subtree can be displayed on a separate window; the subtree can be replaced with an icon; the icon can be restored to the subtree; and keywords contained in the subtree can be searched. These operations can be selected from the menu. In the figure, a branch 505 in the middle of the screen is selected with a mouse cursor 504 or the like represented by an arrow, upon which a menu window 503 appears on which selectable operations are displayed. By transferring the mouse cursor 504 to a desired operation in

the menu window 503, the selected operation is carried out.

Although Ward method is selected as a grouping algorithm in Figure 5, the selection menu 501 can be pulled down to select other algorithm such as nearest neighbor method, furthest neighbor method, group average method, centroid method, median method, flexible method or the like. Similarity or dissimilarity is an index for indicating a degree of similarity between two expression patterns. Such index may be a distance where a shorter distance represents higher similarity, or a value such as a correlation coefficient where a higher value represents higher similarity. The former index is referred to as dissimilarity and the latter as similarity. Although Euclidean distance is selected as dissimilarity in Figure 5, the selection menu 502 can be pulled down to select other types of (dis)similarity such as standardized squared Euclidean distance, Mahalanobis' general distance, Minkowsky distance or the like. The combination of grouping algorithm and dissimilarity type must be appropriate. For example, when centroid method, median method or flexible method is selected as the grouping algorithm, only squared Euclidean distance can be selected as dissimilarity.

Figure 6 is a view showing an exemplary screen displayed upon selecting a command "display this subtree on a separate window" from the menu shown in Figure 5. The selected subtree is rescaled and redisplayed according to the

length from the root to the leaves. This display technique will allow the user to find more detailed state of the branches of the subtree. According to the present system, the selected subtree can be subjected to clustering again by selecting a grouping algorithm and/or (dis)similarity. For example, clusters distant from each other (such as clusters 401 and 402, and clusters 401 and 403 in Figure 4) resulting from the first clustering can be selected and excluded to see a subtree of interest in more detail. A grouping algorithm and/or (dis)similarity can be selected from the grouping algorithm selection menu 501 and the (dis)similarity selection menu 502.

Figure 7 is a view showing an exemplary screen displayed upon selecting a command "replace this subtree with icon" from the menu shown in Figure 5. The subtree 505 can be replaced with an icon 701, by which a global state of the dendrogram can readily be observed. For example, gene groups with similar functions or gene groups with little expression observed can be assembled as a single icon.

Figure 8 is a view showing an exemplary screen displayed upon selecting a command "search for keyword contained in this subtree" from the menu shown in Figure 5. Among genes contained in the selected subtree, genes having gene information with a predetermined keywords are counted and the results are displayed as search results 801. When a keyword 802 is selected from the search results 801 with a

mouse cursor **804** or the like, genes with this keyword **802** (in the figure, "ribosomal") are marked on the dendrogram with marks **803** or the like. By doing so, types of genes assembled in the subtree can readily be known. When the grouping is found to be failed, another grouping algorithm or (dis)similarity can be selected for another clustering. This would aid selection of more appropriate clustering method.

According to the present invention, an analysis can be made effectively on a produced dendrogram.

Thus, a method for displaying a dendrogram according to the present invention comprises the steps of: clustering a plurality of types of biopolymers based on a set of data obtained by experiments of the plurality of biopolymers under different conditions, and displaying the results thereof in a dendrogram format; selecting a subtree in the dendrogram; and displaying the selected subtree on a separate window.

The present invention may comprise the steps of: designating a different clustering method for the biopolymers included in the subtree displayed on the separate window; and clustering the biopolymers included in the subtree again according to the designated clustering method, and displaying the results thereof in a dendrogram format.

Furthermore, a method for displaying a dendrogram according to the present invention comprises the steps of: clustering a plurality of types of biopolymers based on a set of data obtained by experiments of the plurality of

biopolymers under different conditions, and displaying the results thereof in a dendrogram format; selecting a subtree in the dendrogram; and replacing the selected subtree with an icon.

If necessary, the method may further comprise a step of restoring the subtree icon to the original dendrogram subtree format.

A method for displaying a dendrogram according to the present invention comprises the steps of: clustering a plurality of types of biopolymers based on a set of data obtained by experiments of the plurality of biopolymers under different conditions, and displaying the results thereof in a dendrogram format; selecting a subtree in the dendrogram; and from the biopolymers included in the selected subtree, counting and displaying the number of biopolymers containing in their biopolymer information a keyword from a keyword dictionary file.

A method for displaying a dendrogram according to the present invention comprises the steps of: clustering a plurality of types of biopolymers based on a set of data obtained by experiments of the plurality of biopolymers under different conditions, and displaying the results thereof in a dendrogram format; selecting a subtree in the dendrogram; designating a keyword; and displaying a location of a biopolymer in the dendrogram, which includes the designated keyword in its biopolymer information.

According to the above-described methods, the biopolymers may be cDNAs, RNAs, DNA fragments or genes.

A system for displaying a dendrogram according to the present invention comprises: a clustering processor for clustering a plurality of types of biopolymers based on a set of data obtained by experiments of the plurality of biopolymers under different conditions, and analyzing the results thereof to display them in a dendrogram format; a display section for displaying the dendrogram; input means; and a keyword dictionary file for storing keywords of biopolymer information. The input means may be a keyboard or a mouse which is used for selecting a branch in the dendrogram, selecting a clustering method and the like. The keyword dictionary file may be used to evaluate whether the results of clustering have turned out to be successful.

This system for displaying a dendrogram may have a function of displaying a subtree selected by the input means on a separate window. Alternatively, the system may have a function of designating a different clustering method for the subtree displayed on the separate window to cluster the biopolymers included in the subtree again according to the designated clustering method, and displaying the results thereof in a dendrogram format.

The system for displaying a dendrogram may have a function of replacing the subtree selected by the input means with an icon, and a function of restoring the subtree icon to the original subtree in the dendrogram format.

The system for displaying a dendrogram may have a function of counting and displaying the number of biopolymers containing in their biopolymer information a keyword from a keyword dictionary file, and/or a function of displaying a location of a biopolymer in the dendrogram, which includes the designated keyword.

According to the system for displaying a dendrogram of the invention, the biopolymers may be DNAs, RNAs, DNA fragments or genes.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a diagram showing an exemplary display of results of a standard clustering analysis.

Figure 2 is a diagram showing an example of difference between clustering methods.

Figure 3 is a diagram showing exemplary dendrograms with normalized distance (dissimilarity) obtained by different clustering methods.

Figure 4 is a diagram showing an exemplary dendrogram including a gene group with similar expression patterns.

Figure 5 is a view showing an exemplary display screen according to a dendrogram displaying system of the invention.

Figure 6 is a view showing another exemplary display screen according to a dendrogram displaying system of the invention.

Figure 7 is a view showing yet another exemplary display screen according to a dendrogram displaying system of the invention.

Figure 8 is a view showing still yet another exemplary display screen according to a dendrogram displaying system of the invention.

Figure 9 is a schematic view showing an exemplary configuration of a dendrogram displaying system of the invention.

Figure 10 is a diagram showing exemplary gene expression pattern data.

Figure 11 is a diagram showing an exemplary gene information structure.

Figure 12 is a diagram showing an exemplary cluster structure.

Figure 13 is a diagram showing an example for generating a cluster tree structure.

Figure 14 is a diagram showing an exemplary array for storing distances between clusters.

Figure 15 is a diagram showing an exemplary array for storing root nodes of respective windows.

Figure 16 is a diagram showing an example of a structure for storing a query of search and its results.

Figure 17 is a flowchart showing a general process of the present system.

Figure 18 is a flowchart showing a process of reading out gene data.

Figure 19 is a flowchart showing a process for clustering analysis.

Figure 20 is another flowchart showing a process for clustering analysis.

Figure 21 is a flowchart showing a process for replacement/restoration of icon.

Figure 22 is a flowchart showing a process of searching in gene information

Figure 23 is a flowchart showing a process of searching for a keyword (Process A).

Figure 24 is a flowchart showing a process of reading out gene data of a subtree.

Figure 25 is a flowchart showing a process of generating a new cluster for a leaf of a subtree (Process B).

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Hereinafter, the present invention will be described by way of examples with reference to the accompanying drawings. Although genes are exemplified as a subject of clustering in the following examples, the present invention is not limited thereto and is also applicable to other general biopolymers such as cDNAs, RNAs and DNA fragments.

Figure 9 is a schematic view showing a configuration of an exemplary system for displaying a dendrogram according to the invention. The system is provided with gene data 901 for storing gene information and gene expression patterns, a clustering processor 902 for clustering based on the gene

expression patterns, and analyzing and displaying the results in a dendrogram format, a display device **903** on which the dendrogram is displayed, input means such as a keyboard **904** and a mouse **905** for selecting a branch in the dendrogram or for selecting a clustering method, and a keyword dictionary file **906** for storing keywords of gene information to provide means for evaluating whether the results of clustering are in a user's desired form. The clustering processor **902** is realized with a computer and a program thereof. In stead of the storage medium **901**, gene data can be acquired from a database managed by a remote server computer communicating with the system via a network or the like.

Figure **10** is a schematic view showing a specific structure of gene expression pattern data stored in the gene data **901**. According to the present algorithm, the data is stored as a two-dimensional array. Specifically, numerated data of an expression level (brightness of hybridized fluorescent signal) of a gene corresponding to gene ID **(id)** under an experiment case **(no)** is stored as **Exp[id][no]**. The results obtained from a DNA chip spotted with **m** numbers of genes at different positions correspond to a single experiment case.

Figure **11** is a diagram showing an example of a gene information structure for storing information of a gene stored in the gene data **901**. The gene structure includes members representing gene ID **(1101)**, ORF of the gene **(1102)**,

name of the gene (1103) and a function of the gene (1104). The example shown in Figure 11 is merely an example, and the gene information structure may include information other than the attributes mentioned in the figure.

Figure 12 is a diagram showing exemplary structures indicating clusters used in the clustering. Each cluster structure corresponds to either a node or a leaf in a dendrogram. Each of the cluster structures is managed in a window unit. Nodes or leaves in the same window are provided with the same window ID (1207). In order to identify nodes or leaves in the same window from each other, each cluster structure is uniquely assigned with a **clusterNo** (1205). There are three types of cluster structures, and the values of type (1201) may be **leaf**, **node** or **icon**.

A leaf-type cluster structure corresponds to a single gene ID (1206), i.e., a single gene. Based on the gene ID, data of the gene information structure can be referred. A node-type cluster structure is generated upon every joining step during the clustering. Based on this node-type cluster, the two clusters that have been joined can be referred to as **left** value (1202) and **right** value (1203), and the distance ((dis)similarity) therebetween is stored as **distance** value (1204). The **left** and **right** values are represented by **clusterNo** (1205). An icon-type cluster structure is generated upon replacing the subtree with an icon to be treated in the same manner as the leaves upon display. An

icon indicating the subtree is provided on the tip of the branch. An actual cluster at the root of the subtree can be referred to from the **left** value (**1202**).

Figure **13** is a diagram showing a data structure of the cluster structures exemplified in Figure **12**. The data structure is generated during the course of the clustering analysis. First, the cluster structures start with only leaf-type structures. Then, as clustering takes place, every two cluster structures are joined together upon which a node-type cluster structure is generated, thereby forming a tree structure. Each node-type cluster structure includes information of **clusterNo** of the two joined child nodes and the distance ((dis)similarity) therebetween. Relative gene information can be referred to based on gene ID registered in the leaf-type cluster structures. If a subtree is replaced with an icon, an icon-type cluster is inserted into the tree to be treated as a leaf (clusters downstream from the icon-type cluster are not displayed). For restoring the icon, clusters upstream and downstream from the icon-type cluster are rejoined).

Figure **14** is a diagram showing an example of an array for storing dissimilarity values (i.e., distances between clusters) during the course of the clustering analysis. As shown in the figure, dissimilarity values are stored as a two-dimensional array **dist[][]**. **clusterNo**(1205) of clusters corresponding to the indices of the two-dimensional array

dist[][] are stored in **clust_idx[]**. Specifically, the value of the dissimilarity **dist[i][j]** indicates a dissimilarity value between clusters whose **clusterNo** are **clust_idx[i]** and **clust_idx[j]**. For example, as can be appreciated from Figure **14**, the value of dissimilarity **dist[3][4]** between **clusterNo**: 9 as **clust_idx[3]** and **clusterNo**: 25 as **clust_idx[4]** is 21.

Figure **15** is a diagram showing an example of an array for storing root nodes of respective windows. Specifically, the **clusterNo** of the cluster at the root node in each display window is stored in an array **RootNode[]**. In the example shown in Figure **15**, the value of **RootNode[1]** is 569, which means that the root node of the dendrogram displayed on a display window corresponding to window ID:1 is a cluster of **clusterNo**: 569. Similarly, since the value of **RootNode[2]** is 312, the root node of the dendrogram displayed on a display window corresponding to window ID: 2 is a cluster of **clusterNo**: 312.

Figure **16** is a diagram showing an example of a search structure for storing a query of search and its results. For each keyword registered in the keyword dictionary file **906**, a single structure is generated. When some of synonyms are included in the keywords, they can be processed as a single search target. The search structure includes members such as **keyword** (**1601**) for registering a keyword as a search target, **times** (**1602**) indicating the number of the keyword contained in the subtree, **place** (**1603**) for storing locations of genes

on the dendrogram whose gene information include the keyword. As illustrated in Figure 16, synonyms such as **Rat**, **Mouse** and **Mus** can collectively be registered in the **keyword** member so that these three keywords can be treated as an identical search target.

Figure 17 is a flowchart of a general process of the present system.

First, data is read out from the gene data **901** to the clustering processor **902** (Step **1701**), which will be described later in more detail. Then, various parameters required for carrying out a clustering analysis and displaying results are set (Step **1702**). In the present example, a grouping algorithm, a type of (dis)similarity, and whether or not gene information should be displayed are determined.

Next, a clustering analysis takes place (Step **1703**), and the results thereof are displayed (Step **1704**). Detail of the clustering analysis will be described later. During this clustering analysis, information necessary for displaying a dendrogram is collected and input into cluster structures. The results of the analysis are displayed based on these cluster structures and the information of **RootNode[]** indicating the **clusterNo** of the root nodes on respective window. When the cluster structure is of an icon-type, it is processed as a leaf, and an icon representing a subtree is provided at the tip of the branch.

When the subtree in the displayed dendrogram should be

simplified as an icon, or when the icon is to be restored to the original subtree, the following process is conducted (Step 1705). A branch in the dendrogram is selected with a mouse (Step 1706), and the corresponding subtree is replaced with the icon or an icon is restored to a subtree (Step 1707). Replacement and restoration processes will be described later in detail. Thereafter, the results of the analysis are displayed again (Step 1704).

When search should be conducted in the displayed dendrogram based on a keyword stored in the keyword dictionary file 906, the following process is carried out (Step 1708). A branch in the dendrogram is selected with a mouse (Step 1709), and search is performed (Step 1710). The detail of the search will be described later. Since information required for display will be stored in search structures by searching process 1710, a search results window is newly generated based on the search structures to display the results (Step 1711). By selecting a keyword in the search results window with a mouse or the like, the location(s) of the keyword on the dendrogram is(are) marked based on the information of the **place** member(s) of the search structures.

When clustering based on another combining algorithm or (dis)similarity type should be conducted to the displayed dendrogram, the process returns to Step 1702 (Step 1712). Examples of cluster-combining algorithm include nearest

neighbor method, furthest neighbor method, group average method, centroid method, median method, Ward method and flexible method. According to the nearest neighbor method, the furthest neighbor method, the group average method, the Ward method and the flexible method, dissimilarity simply becomes larger as clusters are merged. As two clusters are merged into one, the merged cluster may become closer to or farther from other clusters. The former is referred to as space contraction, and the latter is referred to as space expansion. A case where the distance is unchanged is referred to as space preservation. The nearest neighbor method has a characteristic of space contraction, and the furthest neighbor method and Ward method each have a characteristic of space expansion. The group average method, the centroid method and the median method each have a characteristic of space preservation. The flexible method may have any of the space characteristics depending on parameter settings. There are various types of (dis)similarity. Typical examples of dissimilarity include squared Euclidean distance, standardized squared Euclidean distance, Mahalanobis' general distance and Minkowsky distance. An appropriate dissimilarity can be selected among the above-mentioned distances considering the above-described characteristic and the like.

When a subtree in the displayed dendrogram should be displayed on a separate window (Step 1713), a branch to be

displayed on the separate window is selected in the dendrogram with a mouse (Step **1714**). Then, data corresponding to the selected subtree in the dendrogram is read out (Step **1715**), and the process returns to Step **1702**. Process of reading out data corresponding to the selected subtree will be described later in detail. When no further selection is to be made, the whole process is ended.

Figure **18** is a detailed flowchart of the reading out process **1701** in Figure **17**.

First, the total numbers of genes and experiment cases are registered in **gene_num** and **exp_num**, respectively (Step **1801**). Then, gene information is read out from the gene data **901** to be registered in gene information structures **gene_info[i]** (where **i** = 1, ..., **gene_num**) (Step **1802**). Gene expression data is read out from the gene data **901** to be registered in **Exp[i][j]** (where **i** = 1, ..., **gene_num**, and **j** = 1, ..., **exp_num**) (Step **1803**). Then, **gene_num** is input into **leaf_num** indicating the total number of leaves in the dendrogram (Step **1804**).

Next, leaf-type cluster structures are generated as initial values. The **leaf_num** number of cluster structures are generated. And for **i** = 1, ..., **leaf_num**, **type** member, **clusterNo**, **geneID** and **windowID** are set to **leaf**, **i**, **i** and 1 respectively (Step **1805**). Then, keywords stored in the keyword dictionary file **906** are read out. For each keyword, a search structure is generated, and the keyword is

registered as **search[].keyword** (Step **1806**). The total number of keywords is substituted for **key_num** (Step **1807**). **wid** representing window ID is set to 1 (Step **1808**), and the process is ended.

Figures **19** and **20** are detailed flowcharts of clustering analysis process **1703** in Figure **17**.

Dissimilarity between expression levels of genes displayed on the window whose **window ID** corresponds to **wid** is calculated. Dissimilarity between genes of **clusterNo i** and **clusterNo j** is registered as **dist[i][j]** (Step **1901**). According to the present algorithm, **clusterNo** is sequentially assigned every time a cluster is generated starting from 1. Accordingly, for a next cluster to be generated, **leaf_num+1** is substituted for **newclusterNo** as the number of the next cluster (Step **1902**). As array information for storing distances (dissimilarity) between clusters, **leaf_num** is substituted for **all_clust** representing the number of clusters to be joined, and for **i** = 1, ..., **leaf_num**, **i** is substituted for **cluster_idx[i]** for initialization. The number of the clusters to be joined (**all_clust**) is evaluated as to whether or not it equals to 1. When it does not equal to 1, the following processes are repeated until it equals to 1 (Step **1905**).

First, based on the previously determined distance (dissimilarity) between clusters, clusters to be joined next are determined. For **i** < **j** and **i**, **j** = 1, 2, ..., **all_clust**, a

minimum value of **dist[i][j]**, and **i** and **j** that give the minimum value are obtained to substitute for **d_min**, **i_min** and **j_min**, respectively. Clusters to be joined next are clusters of **clusterNo** represented by **cluster_idx[i_min]** and **cluster_idx[j_min]**. A cluster is newly generated, and **type**, **left**, **right**, **distance**, **clusterNo** and **windowID** are set to **node**, **cluster_idx[i_min]**, **cluster_idx[j_min]**, **d_min**, **newclusterNo** and **wid**, respectively (Step **1907**). Which one of the clusters should be assigned as **left** member and the other as **right** member may be determined by providing a predetermined criterion such as comparison of expression levels.

Then, information of the array storing distances between clusters is updated. First, a distance ((dis)similarity) between a newly generated cluster and other cluster is calculated and overwritten on a location of array **dist[][]** where a distance between a cluster corresponding to **i_min** and other cluster is stored. For **i** = 1, 2, ..., **i_min** - 1, dissimilarity between the newly generated cluster and a cluster whose **clusterNo** corresponds to **cluster_idx[i]** is registered in **dist[i][i_min]** (Step **2001**). For **j** = **i_min+1**, ..., **J_min-1**, **j_min+1**, ..., **all_clust**, dissimilarity between the newly generated cluster and a cluster corresponding to **cluster_idx[j]** is registered as **dist[i_min][j]** (Step **2002**).

Next, information relative to **j_min** is deleted and all of the array data following **j_min** is shifted forward. For **i**

= j_min, ..., all_clust-1, clust_idx[i+1] is substituted for clust_idx[i] (Step 2003). Then, for i and j that satisfy i < j and i, j = j_min, ..., all_clust, dist[i+1][j] is substituted for dist[i][j] (Step 2004). Thereafter, for i and j that satisfy i < j, i = 1, ..., all_clust-1 and j = j_min, ..., all_clust-1, dist[i][j+1] is substituted for dist[i][j] (Step 2005).

Finally, 1 is subtracted from all_clust indicating the number of clusters to be joined (Step 2006). NewclusterNo indicating clusterNo assigned to a new cluster structure is added with 1 (Step 2007).

The above-described process is repeated until all_clust becomes 1. When all_clust becomes 1, cluster_idx[1] indicating clusterNo of a root node of the present window is substituted for RootNode[wid] (Step 1908) and the process is ended.

Figure 21 is a detailed flowchart of process 1707 in Figure 17 for replacement and restoration of an icon.

Clusters corresponding to both ends of the branch selected at Step 1706 are registered. The downstream (leaf side) cluster is substituted for childClust and the upstream (root side) cluster is substituted for parentClust (Steps 2101 and 2102). Then, a new icon-type cluster is generated and inserted between the childClust and parentClust. Specifically, a cluster is generated, where type, left, clusterNo and windowID are set to icon, childClust.clusterNo,

newclusterNo and **wid**, respectively (Step **2103**). To re-link the pointer, **clusterNo** of **childClust** registered in either **parentClust.left** or **parentClust.right** is replaced with **newclusterNo** (Step **2104**). As the total number of clusters will be increased by one, **newclusterNo** is added with 1 to indicate **clusterNo** assigned to a new cluster structure (Step **2105**). Then, the process is ended.

When restoration of the subtree icon is selected from the menu, first, clusters corresponding to both ends of the branch selected at Step **1706** in Figure **17** are registered. The cluster of the icon downstream from (on the leaf side of) the branch selected at Step **1706** and the cluster at the parent node of the icon are substituted for **iconClust** and **parentClust**, respectively (Steps **2101** and **2106**). The pointer linking the cluster of the icon is re-linked to the clusters of the subtree, and the cluster of the icon is deleted. Specifically, **clusterNo** of **iconClust** registered in either **parentClust.left** or **parentClust.right** is changed into **iconClust.left** (Step **2107**). Then, **iconClust** is deleted (Step **2108**) and the process is ended.

Figure **22** is a detailed flowchart of searching process **1710** in Figure **17**.

First, **clusterNo** of a cluster at a root node of a subtree downstream from the selected branch is substituted for **clustNo** (Step **2201**). Then, **leafNo** indicating an index assigned from the beginning of the leaves in the subtree is

initialized to 1 (Step **2202**). For **i** = 1, ..., **key_num**, **search[i].times** and **search[i].place** are initialized to 0 and null, respectively (Step **2203**). Then, treewalk is recursively performed on the cluster tree to search for a gene having the keyword designated in **search** (Process A) (Step **2205**). Here, **clustNo** and **leafNo** are given as arguments. The detail of keyword searching process will be described later in detail. After Process A, the search results are input into the search structure and the process is ended.

Figure **23** is a detailed flowchart of keyword searching process (Process A) in Figure **22**.

The given arguments **clustNo** and **leafNo** are substituted for **clustNo** and **leafNo**, respectively (Step **2300**). The cluster corresponding to **clusterNo** is substituted for **targetClust** (Step **2301**). A counter **i** for keyword search is set to 0 (Step **2302**).

Then, **targetCluster.type** is evaluated as to whether it is **leaf** or not (Step **2303**). When it is **leaf**, the following process is repeated until gene information corresponding to **leaf** is completely compared with the keyword read out from the keyword dictionary file. In other words, the process is repeated until **i** becomes **key_num** (Step **2304**). First, the attribute of gene information structure **gene_info** corresponding to **targetClust.geneID** is evaluated as to inclusion of keyword **search[i].keyword** (Step **2305**). If the keyword is included, **search[i].times**, which indicates the

number of detection of the keyword (**search[i].keyword**) in the subtree, is increased by 1. Then, **leafNo** of the detected location is registered in **search[i].place** indicating the index of the detected location in the subtree (Step **2307**). The counter **i** for keyword search is increased by 1 and the process returns to Step **2304**. When **i** becomes **key_num** at Step **2304**, i.e., when entire keywords are completely compared, **leafNo** as an index of the subtree is increased by 1 (Step **2309**) and the process is ended.

When **targetCluster.type** is not **leaf** at Step **2303**, a child node is traced. First, **targetClust.left** is substituted for **clustNo** (Step **2310**), and the keyword searching process (Process A) is performed on left child node using **clustNo** and **leftNo** as arguments (Step **2311**). When **targetCluster.type** is **icon**, **targetCluster.right** has no child node (Step **2312**) and thus the process is ended. When **targetCluster.type** is not **icon** at Step **2312**, the cluster is of a **node** type. Thus, **targetClust.right** is substituted for **clustNo** (Step **2313**), and keyword searching process (Process A) is repeated on the right child node using **clustNo** and **leafNo** as arguments (Step **2314**) and the process is ended.

Figure **24** is a detailed flowchart of process **1715** in Figure **17**, for reading out gene data of the subtree.

Since a subtree is newly read out and a window is newly generated, **wid** indicating a new window ID is increased by 1 (Step **2401**). In addition, **leaf_num** indicating the total

number of leaves in the dendrogram is initialized to 0 (Step **2402**). Then, **clusterNo** of a cluster at the root node of the subtree downstream from the selected branch is substituted for **clusterNo** (Step **2403**). Finally, process of generating new cluster (Process B) is performed on the leaf-type cluster of the subtree (Step **2404**). For this process, **clustNo** indicating the present cluster is given as an argument. This process will be described later in detail. After reading out all leaves and generating all clusters corresponding to the leaves, the process is ended.

Figure **25** is a detailed flowchart of process **2404** in Figure **24**, for generating a new cluster corresponding to a leaf in the subtree.

The given argument **clustNo** is registered as **clustNo**, and the cluster indicated by the given **clustNo** is set as **targetClust** (Steps **2501** and **2502**). Then, **targetCluster.type** is evaluated as to whether it is **leaf** or not (Step **2503**). If it is **leaf**, **leaf_num** as a counter of the number of leaves of the subtree is increased by 1 (Step **2504**). Then, a leaf-type cluster structure is generated as an initial value of the new window. Specifically, a cluster is generated where **type**, **clusterNo**, **geneID** and **windowID** are set to **leaf**, **leaf_num**, **targetCluster.geneID** and **wid**, respectively, thereby ending the process (Step **2505**).

When **targetCluster.type** is not **leaf** at Step **2503**, a child node is traced. First, **targetClust.left** is substituted

for **clustNo** (Step **2506**), and a cluster is newly generated again using **clustNo** as an argument (Process B) (Step **2507**). When **targetCluster.type** is **icon**, **targetCluster.right** has no child node, and thus the process is ended (Step **2508**). When **targetCluster.type** is not **icon** at Step **2508**, the cluster is of a node type. Accordingly, **targetClust.right** is substituted for **clustNo** (Step **2509**), and a new cluster generating process (Process B) is repeated for the right child node using **clustNo** as an argument and the process is ended (Step **2510**).

Herein, the result of the analysis is displayed only on a display device. However, the results can be printed out with a multicolor printer. According to the present invention, the idea of display also comprises a printed out display.

According to the present invention, a method for aiding gene expression analysis or the like is provided, where various clustering methods can be applied to a dendrogram, and a subtree can be replaced with an icon or displayed on a separate window.

WHAT IS CLAIMED IS:

1. A method for displaying a dendrogram comprising the steps of:

clustering a plurality of types of biopolymers based on a set of data obtained by experiments of the plurality of biopolymers under different conditions, and displaying the results thereof in a dendrogram format;

selecting a subtree in the dendrogram; and

displaying the selected subtree on a separate window.

2. A method for displaying a dendrogram according to claim 1, further comprising the steps of:

designating a different clustering method for the biopolymers included in the subtree displayed on the separate window; and

clustering the biopolymers included in the subtree again according to the designated clustering method, and displaying the results thereof in a dendrogram format.

3. A method for displaying a dendrogram comprising the steps of:

clustering a plurality of types of biopolymers based on a set of data obtained by experiments of the plurality of biopolymers under different conditions, and displaying the results thereof in a dendrogram format;

selecting a subtree in the dendrogram; and

replacing the selected subtree with an icon.

4. A method for displaying a dendrogram according to claim 3, further comprising a step of restoring the subtree icon to the original dendrogram subtree format.

5. A method for displaying a dendrogram comprising the steps of:

clustering a plurality of types of biopolymers based on a set of data obtained by experiments of the plurality of biopolymers under different conditions, and displaying the results thereof in a dendrogram format;

selecting a subtree in the dendrogram; and

from the biopolymers included in the selected subtree, counting and displaying the number of biopolymers containing in their biopolymer information a keyword from a keyword dictionary file.

6. A method for displaying a dendrogram comprising the steps of:

clustering a plurality of types of biopolymers based on a set of data obtained by experiments of the plurality of biopolymers under different conditions, and displaying the results thereof in a dendrogram format;

selecting a subtree in the dendrogram;

designating a keyword; and

displaying a location of a biopolymer in the dendrogram, which includes the designated keyword in its biopolymer information.

7. A method for displaying a dendrogram according to any one of claims 1 to 6, wherein the biopolymers are cDNAs, RNAs, DNA fragments or genes.

8. A system for displaying a dendrogram comprising:

a clustering processor for clustering a plurality of types of biopolymers based on a set of data obtained by experiments of the plurality of biopolymers under different conditions, and analyzing the results thereof to display them in a dendrogram format;

a display system for displaying the dendrogram;

input means; and

a keyword dictionary file for storing keywords of biopolymer information.

9. A system for displaying a dendrogram according to claim 8, comprising a function of displaying a subtree selected by the input means on a separate window.

10. A system for displaying a dendrogram according to claim 9, comprising a function of designating a different clustering method for the subtree displayed on the separate window to cluster the biopolymers included in the subtree again according to the designated clustering method, and displaying the results thereof in a dendrogram format.

11. A system for displaying a dendrogram according to any one of claims 8 to 10, wherein the system comprises a function of replacing the subtree selected by the input means with an icon, and a function of restoring the subtree icon to the original subtree in the dendrogram format.

12. A system for displaying a dendrogram according to any one of claims 8 to 11, wherein the system comprises a function of counting and displaying the number of biopolymers containing in their biopolymer information a keyword from a keyword dictionary file, and/or a function of displaying a location of a biopolymer in the dendrogram, which includes the designated keyword.

13. A system for displaying a dendrogram according to any one of claims 8 to 12, wherein the biopolymers are cDNAs, RNAs, DNA fragments or genes.

ABSTRACT

The present invention represents a global state of branches in a whole dendrogram as well as detail of states of individual subtrees, to aid focusing of groupings and selection of a clustering method.

The present invention has functions of selecting a branch in a dendrogram, displaying a subtree including the selected branch and its leaves on a separate window, replacing the subtree with an icon, restoring the icon into the original subtree, and collecting and displaying keywords contained in the subtree.

Fig. 1

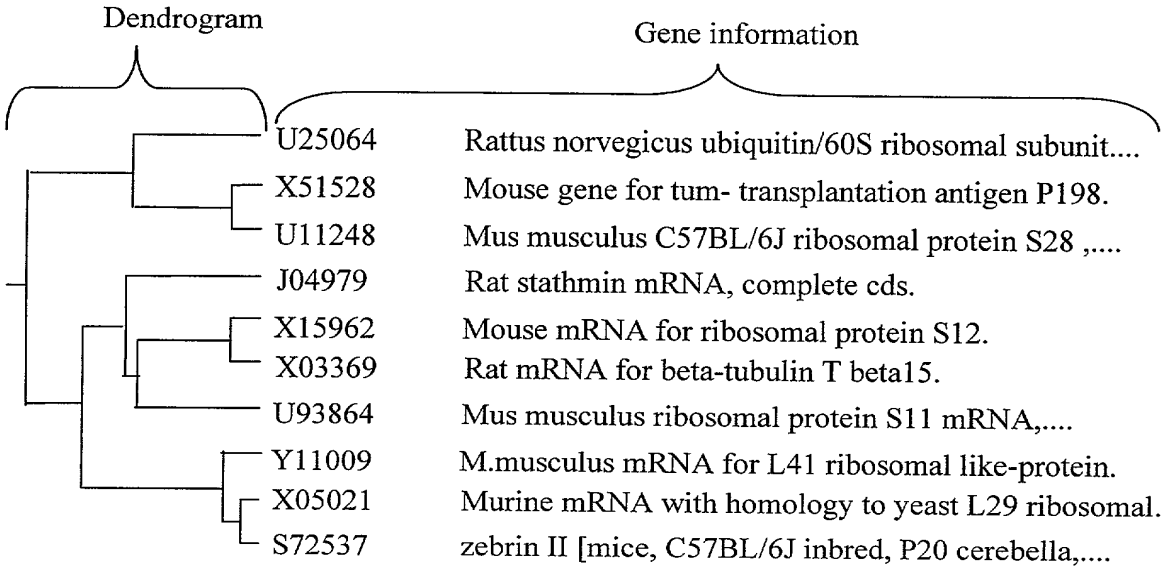

Dendrogram
Gene information
U25064
X51528
U11248
J04979
X15962
X03369
U93864
Y11009
X05021
S72537
Rattus norvegicus ubiquitin/60S ribosomal subunit....
Mouse gene for tum- transplantation antigen P198.
Mus musculus C57BL/6J ribosomal protein S28 ,....
Rat stathmin mRNA, complete cds.
Mouse mRNA for ribosomal protein S12.
Rat mRNA for beta-tubulin T beta15.
Mus musculus ribosomal protein S11 mRNA,....
M.musculus mRNA for L41 ribosomal like-protein.
Murine mRNA with homology to yeast L29 ribosomal.
zebrin II [mice, C57BL/6J inbred, P20 cerebella,....

Fig. 2

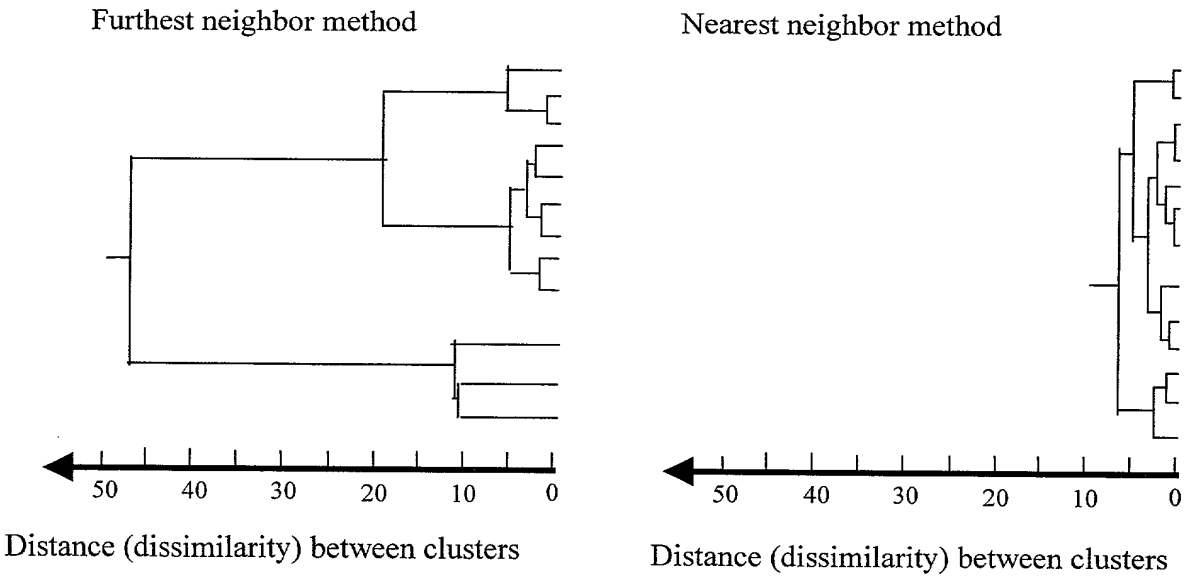

Furthest neighbor method
50
40
30
20
10
0
Distance (dissimilarity) between clusters
Nearest neighbor method
50
40
30
20
10
0
Distance (dissimilarity) between clusters

Fig. 3

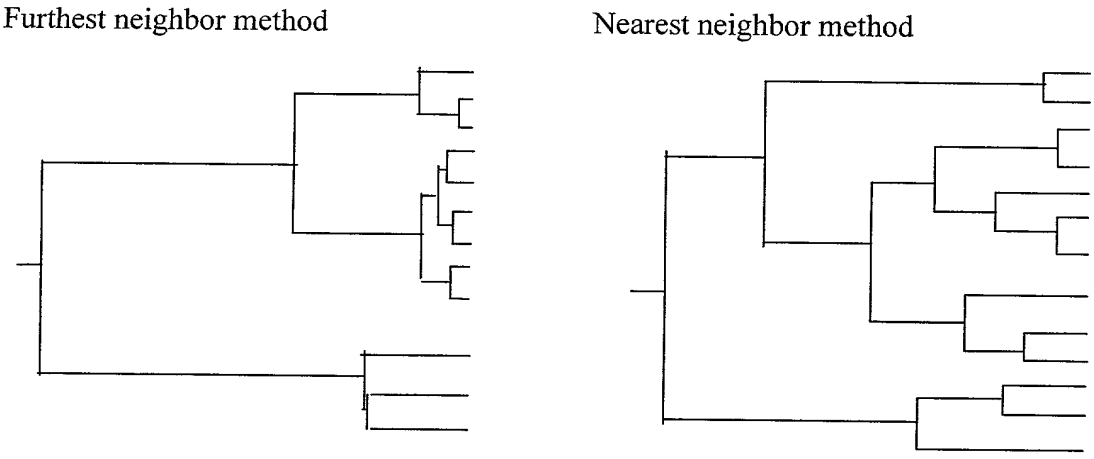

Furthest neighbor method
Nearest neighbor method

Fig. 4

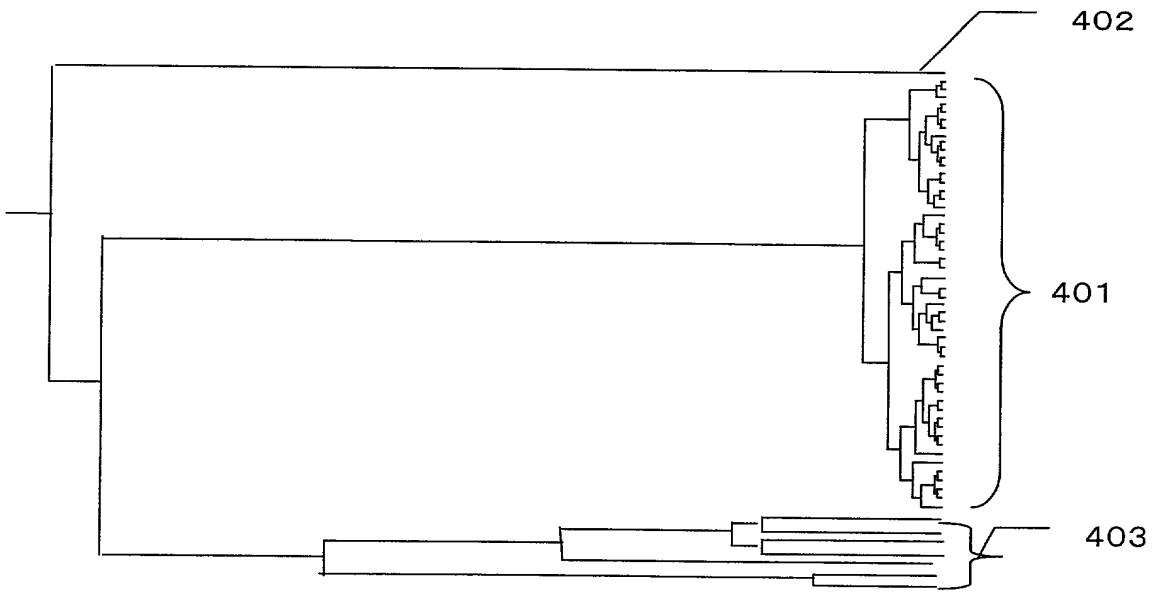

402
401
403

Fig. 5

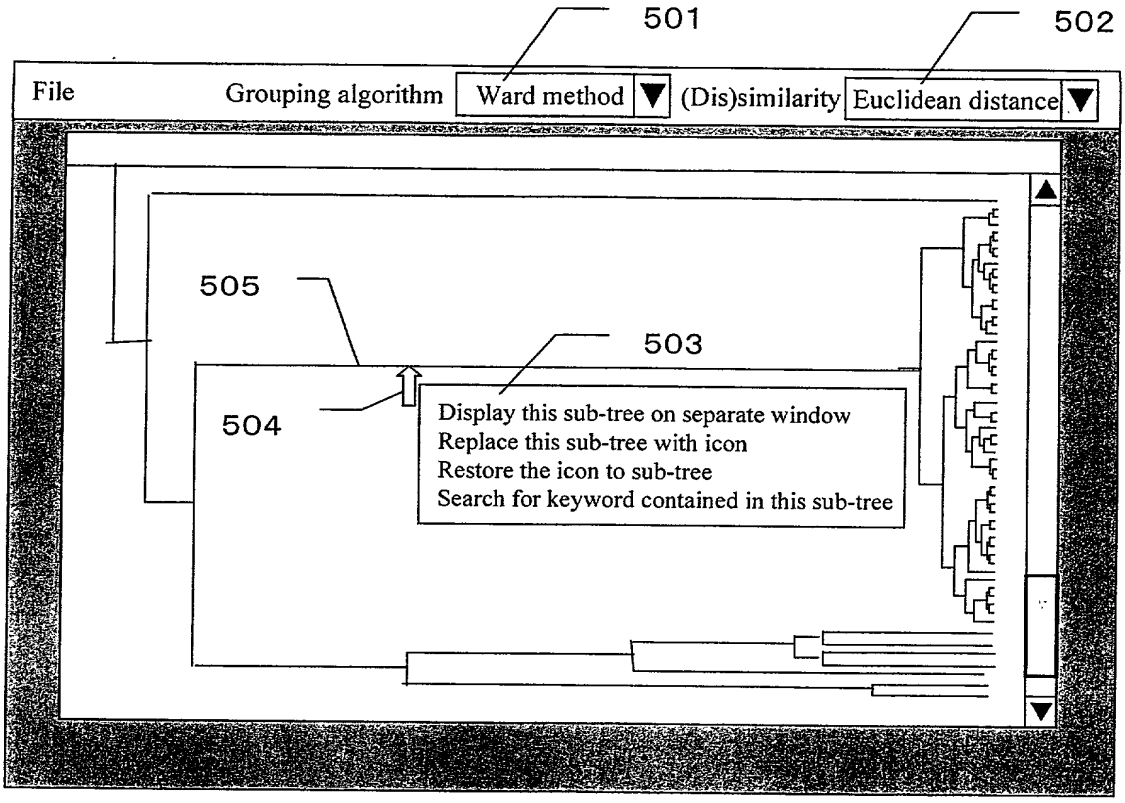

501
502
File
Grouping algorithm
Ward method
(Dis)similarity
Euclidean distance
505
503
504
Display this sub-tree on separate window
Replace this sub-tree with icon
Restore the icon to sub-tree
Search for keyword contained in this sub-tree

Fig. 6

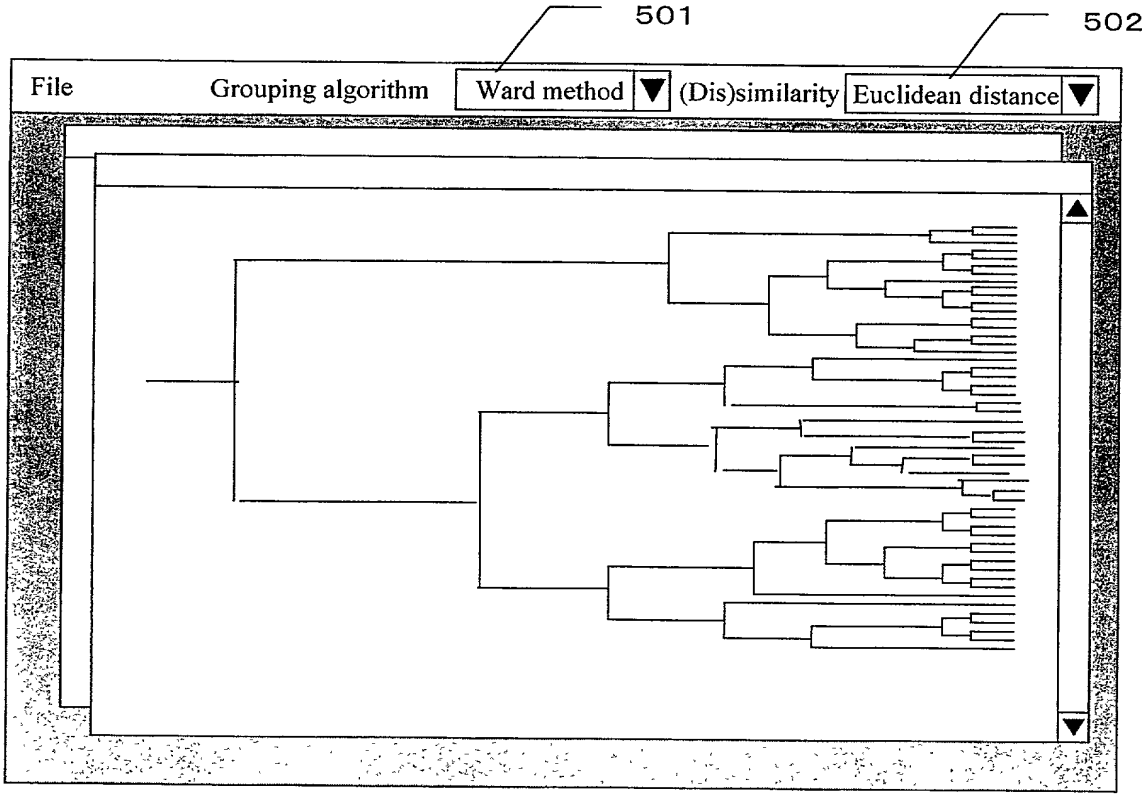

501
502
File
Grouping algorithm
Ward method
(Dis)similarity
Euclidean distance

Fig. 7

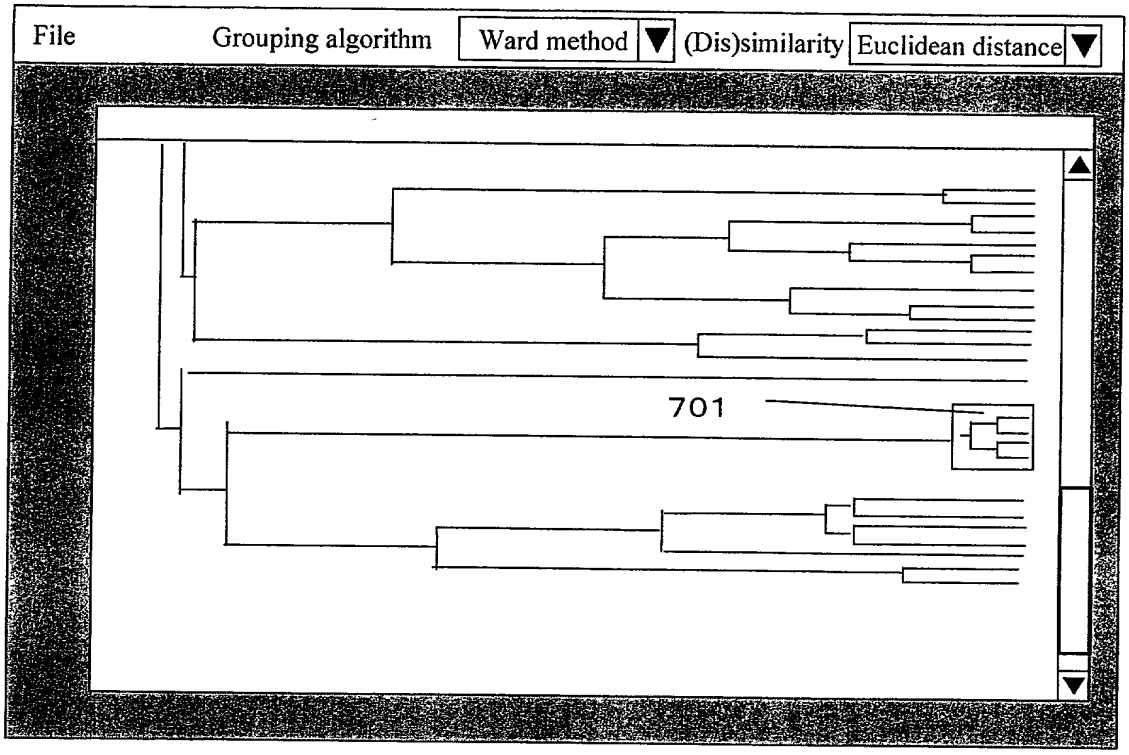

File
Grouping algorithm
Ward method
(Dis)similarity
Euclidean distance
701

Fig. 8

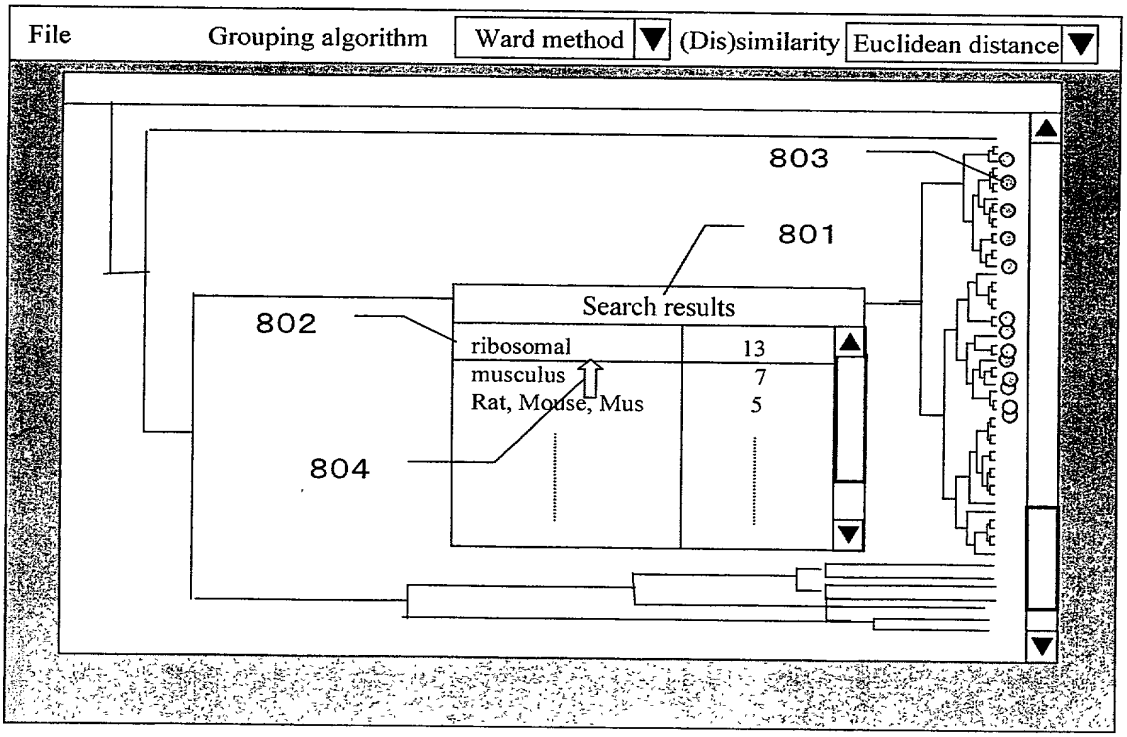

File
Grouping algorithm
Ward method
(Dis)similarity
Euclidean distance
803
801
Search results
802
ribosomal
13
musculus
7
Rat, Mouse, Mus
5
804

Fig. 9

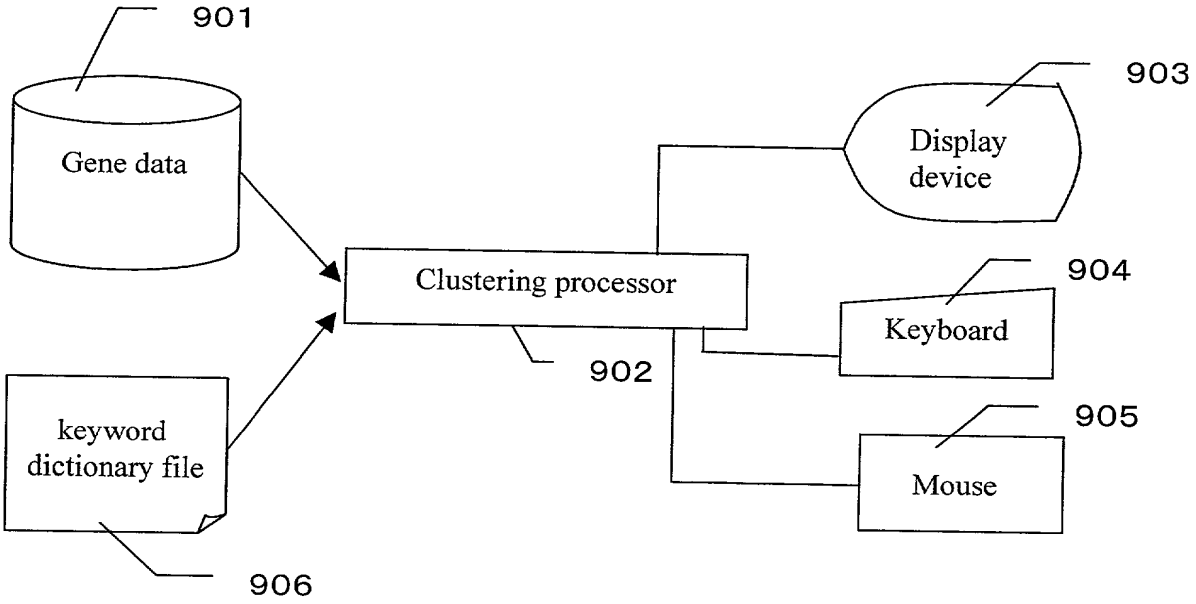

901
Gene data
903
Display device
Clustering processor
904
Keyboard
902
keyword dictionary file
905
Mouse
906

Fig. 10

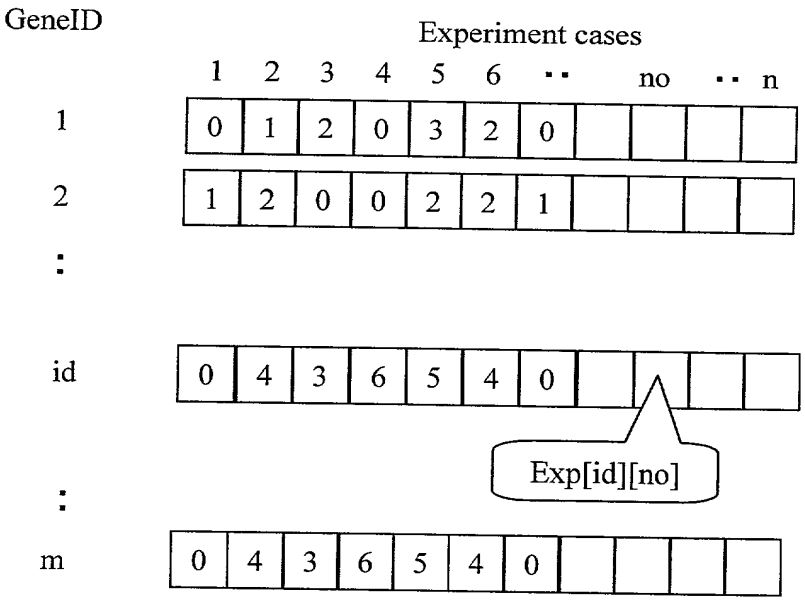

GeneID
Experiment cases
1 2 3 4 5 6 ·· no ·· n
1
0 1 2 0 3 2 0
2
1 2 0 0 2 2 1
:
id
0 4 3 6 5 4 0
Exp[id][no]
:
m
0 4 3 6 5 4 0

Fig. 11

gene_info

	Member	Value
1101	geneID	17
1102	ORF	YBL084C
1103	name	ANAPHASE-PROMOTING COMPLEX SUBUNIT
1104	function	CELL CYCLE

Fig. 12

cluster

	Member	Value
1201	type	leaf
1202	left	
1203	right	
1204	distance	
1205	clusterNo	17
1206	geneID	11
1207	windowID	3

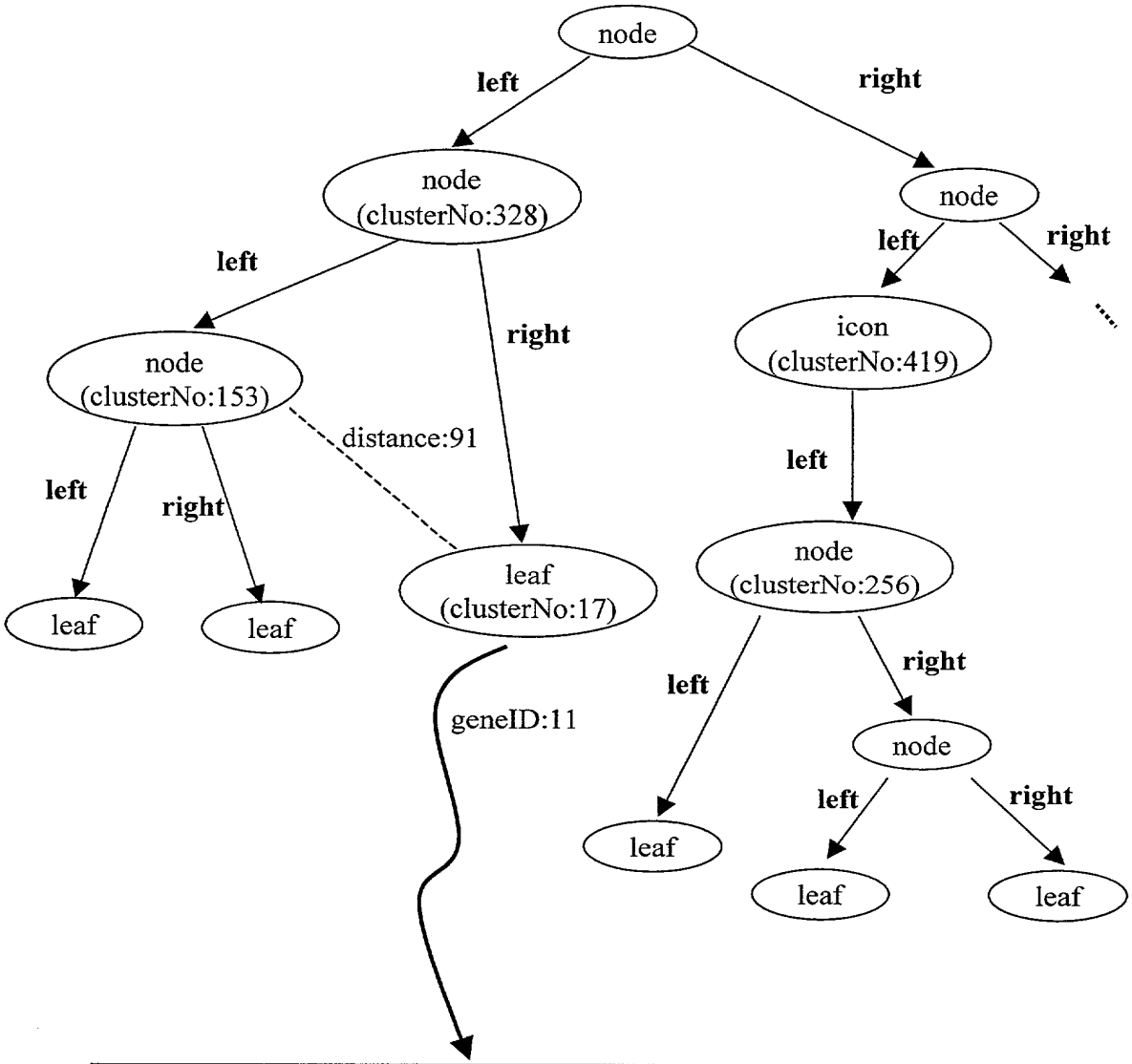
cluster

Member	Value
type	node
left	153
right	17
distance	91
clusterNo	328
geneID	
windowID	3

cluster

Member	Value
type	icon
left	256
right	
distance	
clusterNo	419
geneID	
windowID	3

Fig. 13



geneID	11
ORF	YDR342C
name	HEXOSE PERMEASE
function	TRANSPORT

Fig. 14

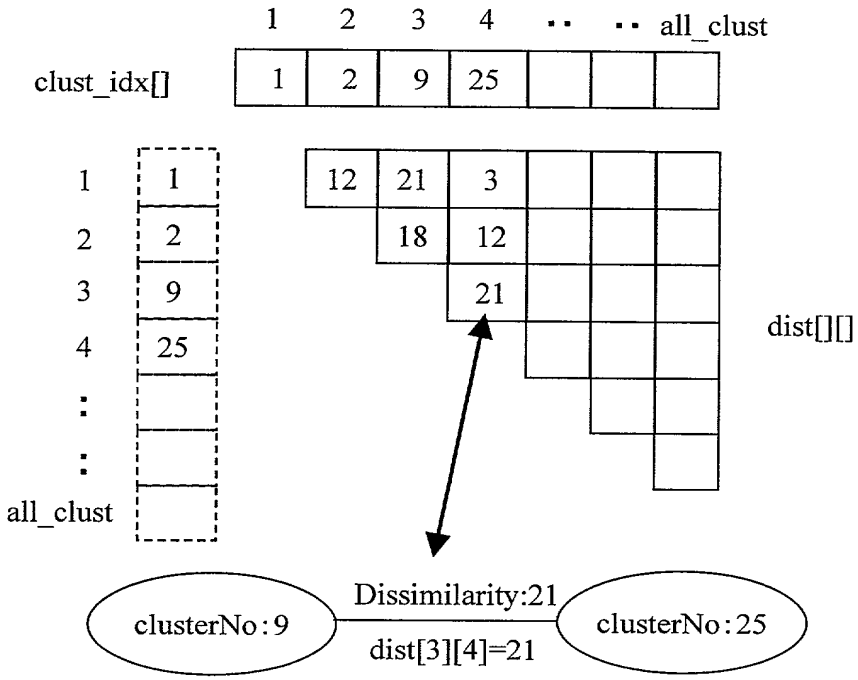

1 2 3 4 .. .. all_clust
clust_idx[]
1 2 9 25
1
2
3
4
:
:
all_clust
1
2
9
25
12 21 3
18 12
21
dist[][]
clusterNo:9
Dissimilarity:21
dist[3][4]=21
clusterNo:25

Fig. 15

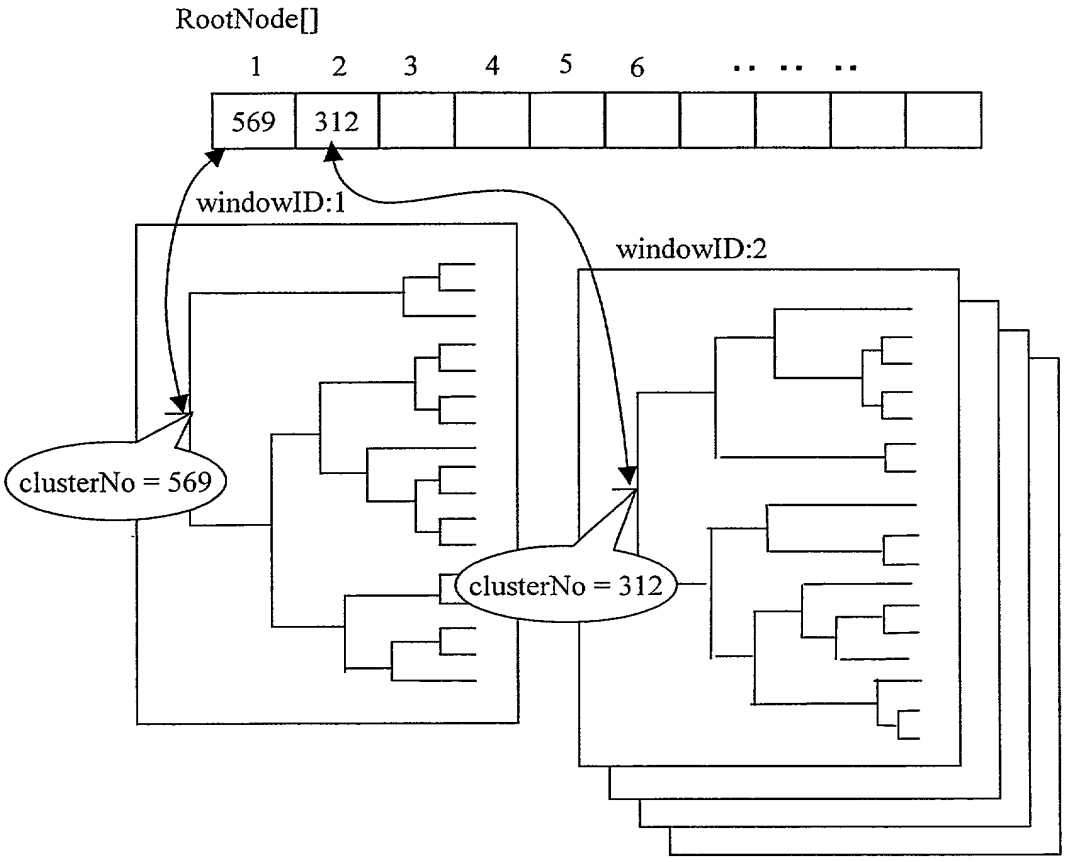

RootNode[]
1
2
3
4
5
6
569
312
windowID:1
windowID:2
clusterNo = 569
clusterNo = 312

Fig. 16

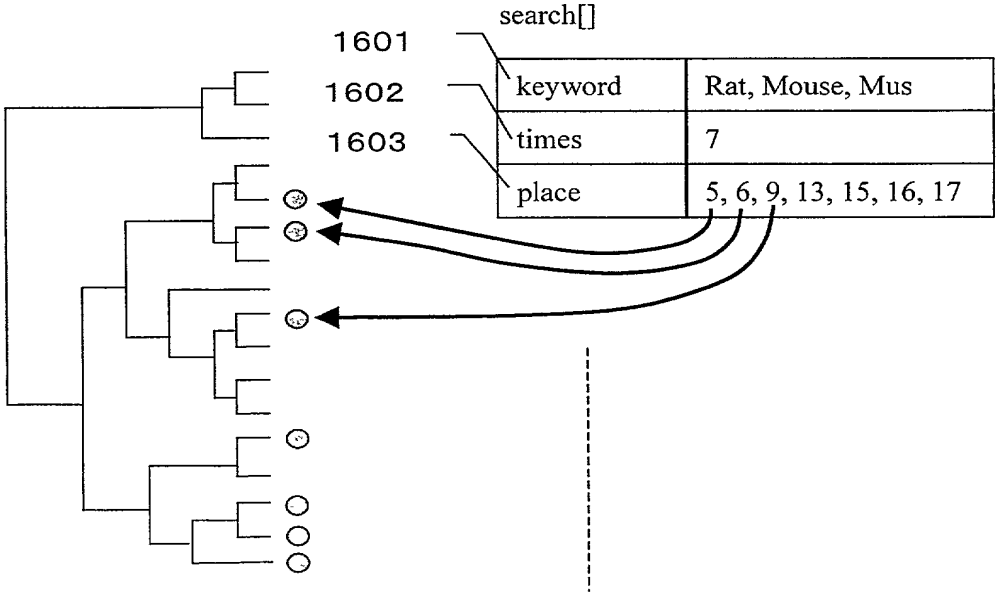

search[]
1601
1602
1603
keyword
Rat, Mouse, Mus
times
7
place
5, 6, 9, 13, 15, 16, 17

Fig. 17

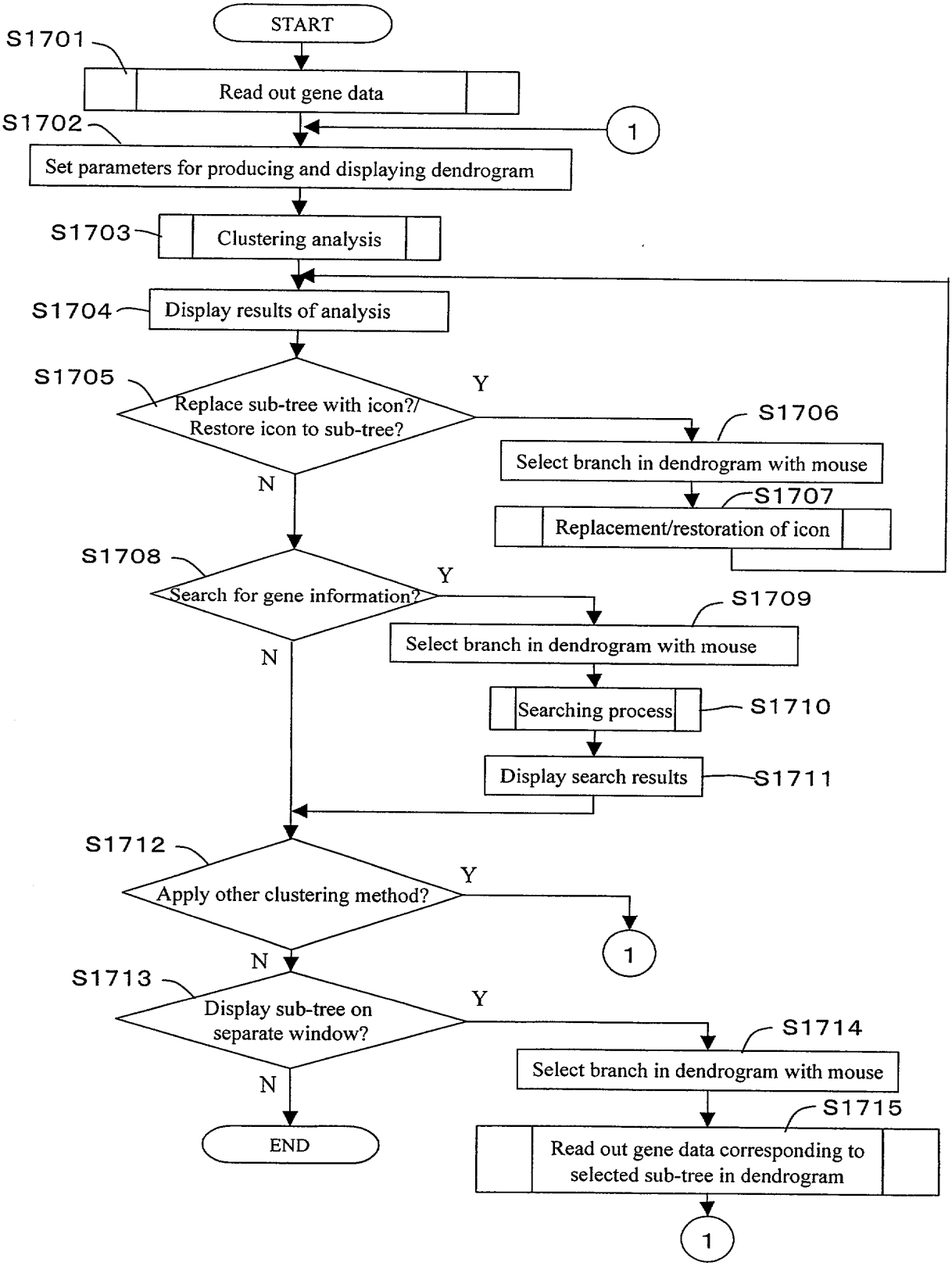

START
S1701
Read out gene data
S1702
1
Set parameters for producing and displaying dendrogram
S1703
Clustering analysis
S1704
Display results of analysis
S1705
Replace sub-tree with icon?/
Restore icon to sub-tree?
Y
S1706
Select branch in dendrogram with mouse
S1707
Replacement/restoration of icon
N
S1708
Search for gene information?
Y
S1709
Select branch in dendrogram with mouse
N
Searching process
S1710
Display search results
S1711
S1712
Apply other clustering method?
Y
1
N
S1713
Display sub-tree on
separate window?
Y
S1714
Select branch in dendrogram with mouse
N
S1715
END
Read out gene data corresponding to
selected sub-tree in dendrogram
1

Fig. 18

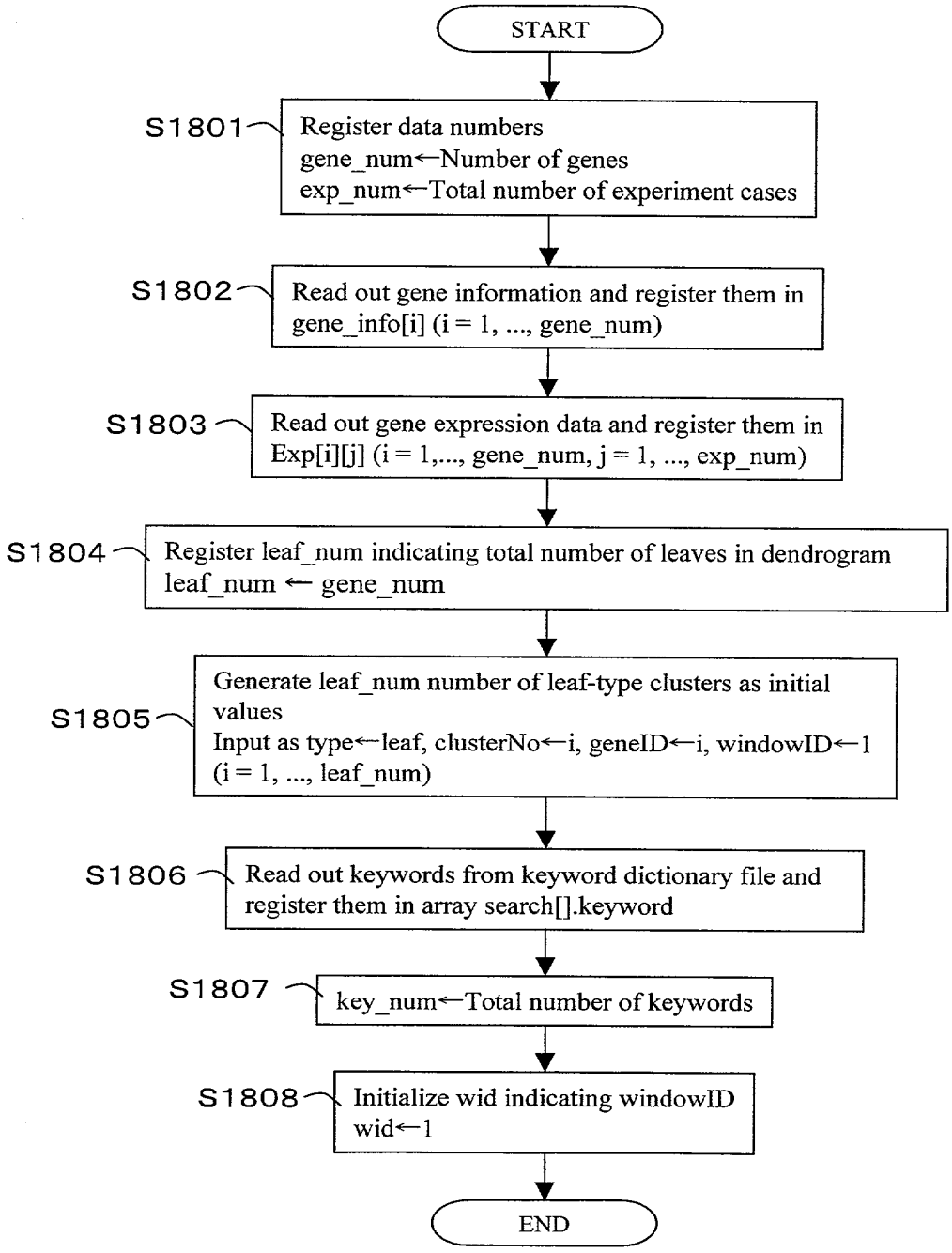

START
S1801
Register data numbers
gene_num←Number of genes
exp_num←Total number of experiment cases
S1802
Read out gene information and register them in
gene_info[i] (i = 1, ..., gene_num)
S1803
Read out gene expression data and register them in
Exp[i][j] (i = 1,..., gene_num, j = 1, ..., exp_num)
S1804
Register leaf_num indicating total number of leaves in dendrogram
leaf_num ← gene_num
S1805
Generate leaf_num number of leaf-type clusters as initial values
Input as type←leaf, clusterNo←i, geneID←i, windowID←1
(i = 1, ..., leaf_num)
S1806
Read out keywords from keyword dictionary file and
register them in array search[].keyword
S1807
key_num←Total number of keywords
S1808
Initialize wid indicating windowID
wid←1
END

Fig. 19

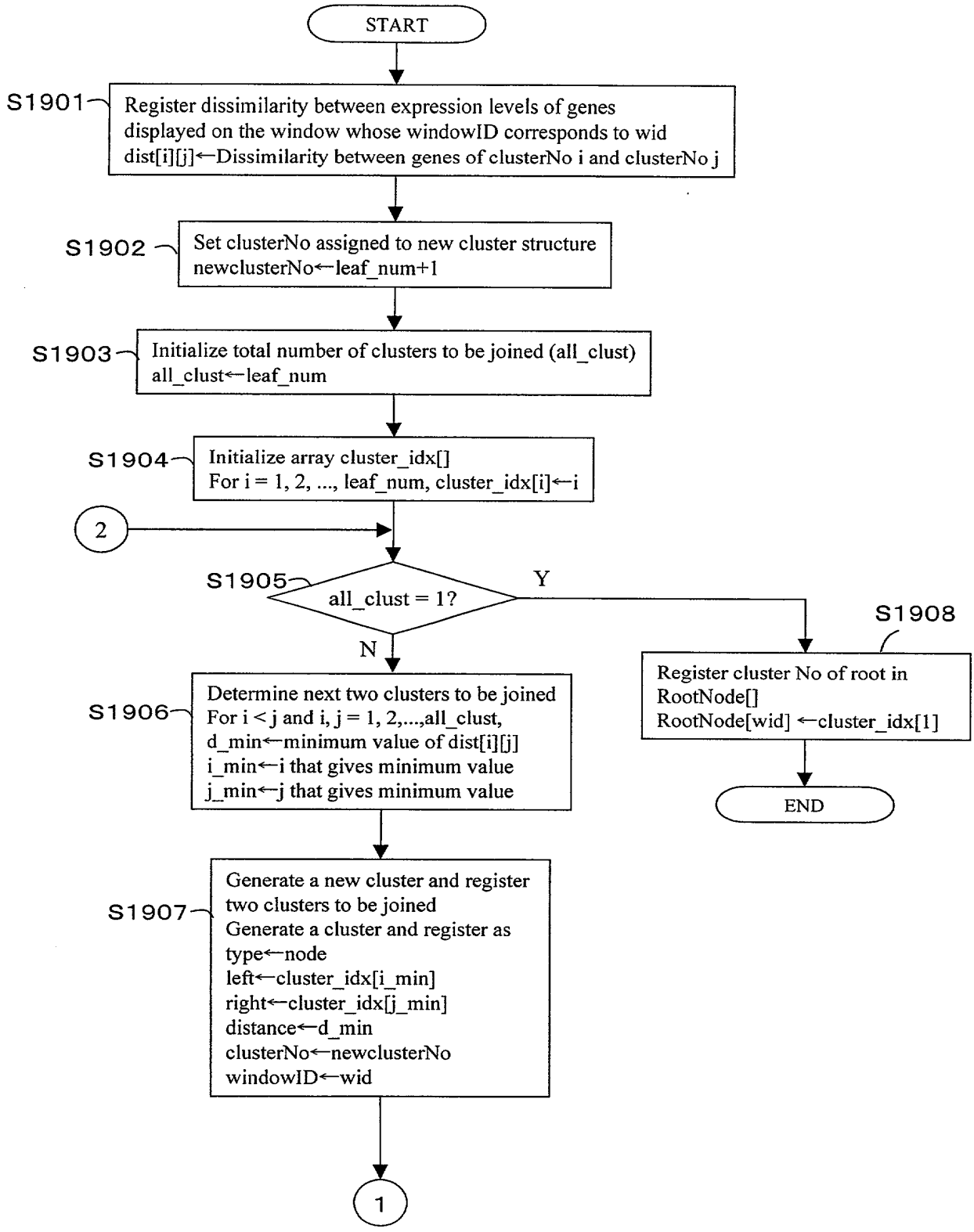

START
S1901
Register dissimilarity between expression levels of genes displayed on the window whose windowID corresponds to wid
dist[i][j]←Dissimilarity between genes of clusterNo i and clusterNo j
S1902
Set clusterNo assigned to new cluster structure
newclusterNo←leaf_num+1
S1903
Initialize total number of clusters to be joined (all_clust)
all_clust←leaf_num
S1904
Initialize array cluster_idx[]
For i = 1, 2, ..., leaf_num, cluster_idx[i]←i
2
S1905
all_clust = 1?
Y
N
S1908
Register cluster No of root in RootNode[]
RootNode[wid] ←cluster_idx[1]
END
S1906
Determine next two clusters to be joined
For i < j and i, j = 1, 2,...,all_clust,
d_min←minimum value of dist[i][j]
i_min←i that gives minimum value
j_min←j that gives minimum value
S1907
Generate a new cluster and register two clusters to be joined
Generate a cluster and register as
type←node
left←cluster_idx[i_min]
right←cluster_idx[j_min]
distance←d_min
clusterNo←newclusterNo
windowID←wid
1

Fig. 20

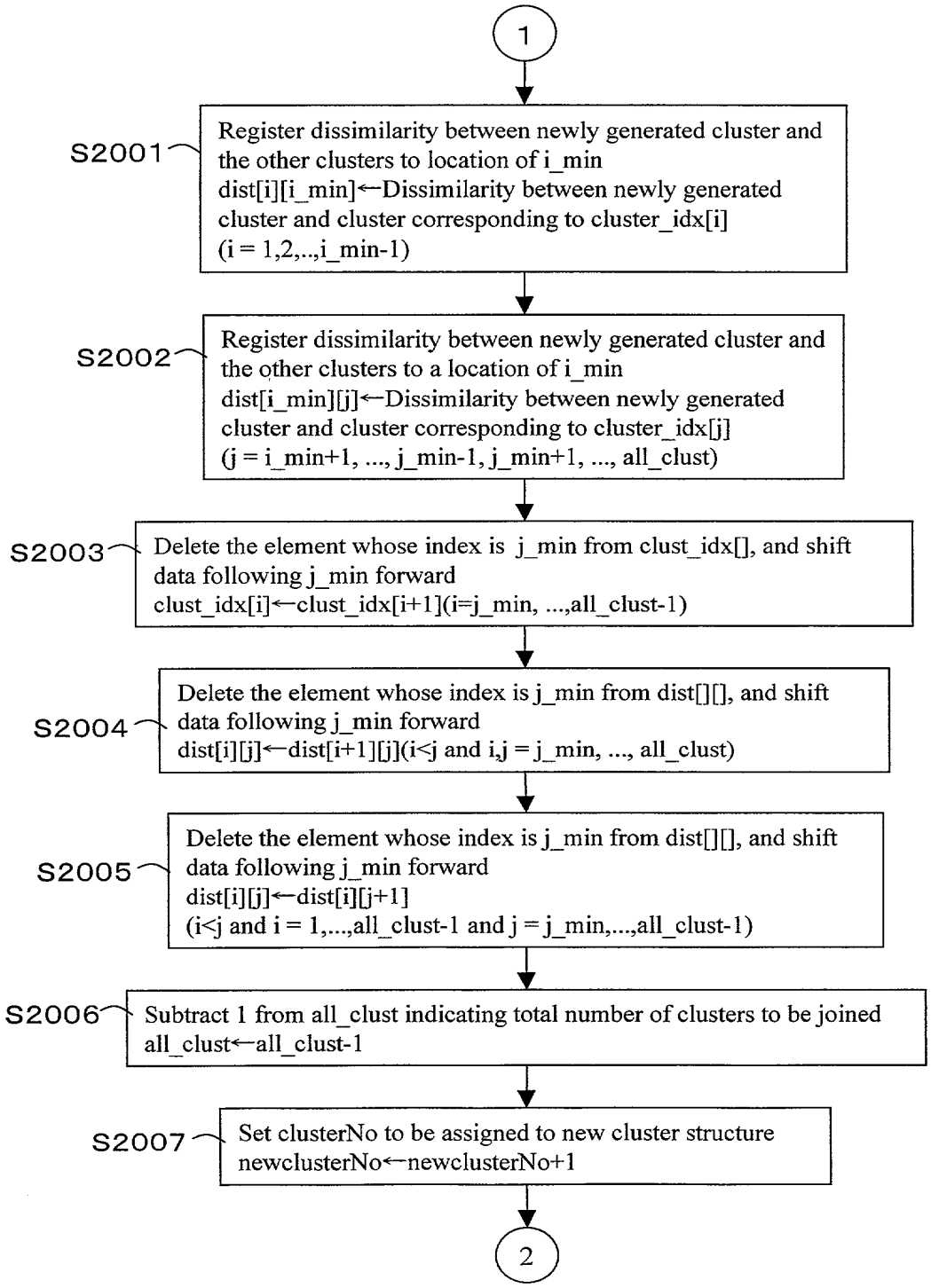

1
S2001
Register dissimilarity between newly generated cluster and the other clusters to location of i_min
dist[i][i_min]←Dissimilarity between newly generated cluster and cluster corresponding to cluster_idx[i]
(i = 1,2,..,i_min-1)
S2002
Register dissimilarity between newly generated cluster and the other clusters to a location of i_min
dist[i_min][j]←Dissimilarity between newly generated cluster and cluster corresponding to cluster_idx[j]
(j = i_min+1, ..., j_min-1, j_min+1, ..., all_clust)
S2003
Delete the element whose index is j_min from clust_idx[], and shift data following j_min forward
clust_idx[i]←clust_idx[i+1](i=j_min, ...,all_clust-1)
S2004
Delete the element whose index is j_min from dist[][], and shift data following j_min forward
dist[i][j]←dist[i+1][j](i<j and i,j = j_min, ..., all_clust)
S2005
Delete the element whose index is j_min from dist[][], and shift data following j_min forward
dist[i][j]←dist[i][j+1]
(i<j and i = 1,...,all_clust-1 and j = j_min,...,all_clust-1)
S2006
Subtract 1 from all_clust indicating total number of clusters to be joined
all_clust←all_clust-1
S2007
Set clusterNo to be assigned to new cluster structure
newclusterNo←newclusterNo+1
2

Fig. 21

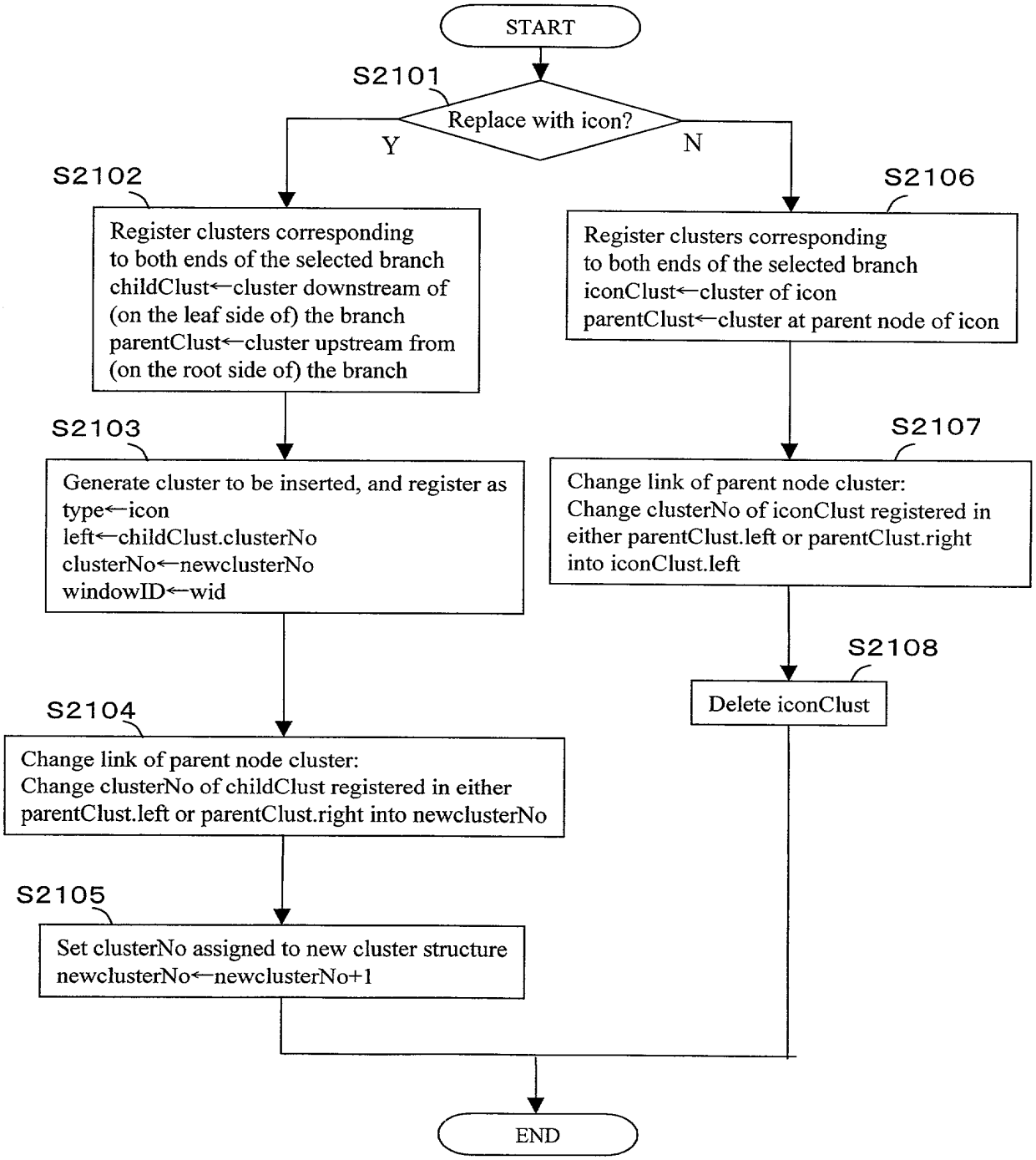

START
S2101
Replace with icon?
Y
N
S2102
Register clusters corresponding to both ends of the selected branch
childClust←cluster downstream of (on the leaf side of) the branch
parentClust←cluster upstream from (on the root side of) the branch
S2103
Generate cluster to be inserted, and register as
type←icon
left←childClust.clusterNo
clusterNo←newclusterNo
windowID←wid
S2104
Change link of parent node cluster:
Change clusterNo of childClust registered in either parentClust.left or parentClust.right into newclusterNo
S2105
Set clusterNo assigned to new cluster structure
newclusterNo←newclusterNo+1
S2106
Register clusters corresponding to both ends of the selected branch
iconClust←cluster of icon
parentClust←cluster at parent node of icon
S2107
Change link of parent node cluster:
Change clusterNo of iconClust registered in either parentClust.left or parentClust.right into iconClust.left
S2108
Delete iconClust
END

Fig. 22

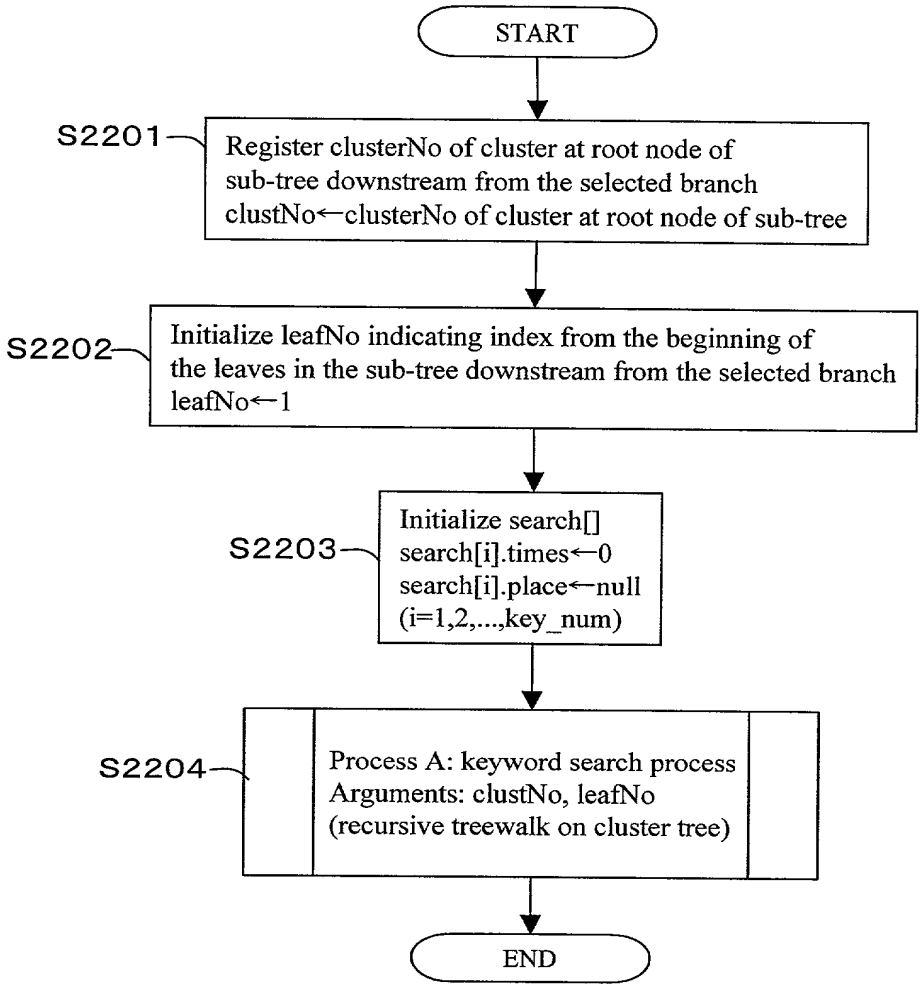

START
S2201
Register clusterNo of cluster at root node of sub-tree downstream from the selected branch
clustNo←clusterNo of cluster at root node of sub-tree
S2202
Initialize leafNo indicating index from the beginning of the leaves in the sub-tree downstream from the selected branch
leafNo←1
S2203
Initialize search[]
search[i].times←0
search[i].place←null
(i=1,2,...,key_num)
S2204
Process A: keyword search process
Arguments: clustNo, leafNo
(recursive treewalk on cluster tree)
END

Fig. 23

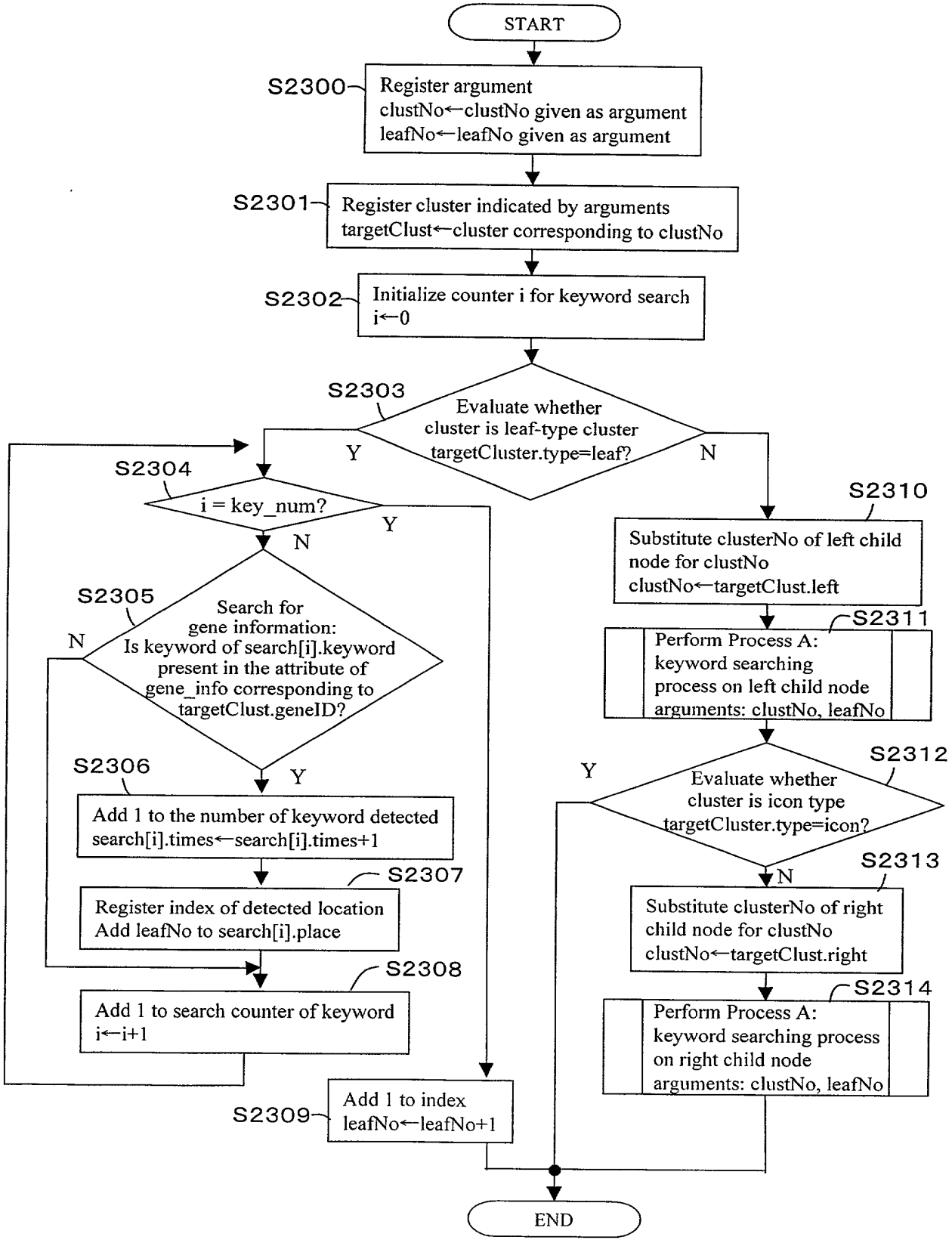

START
S2300
Register argument
clustNo←clustNo given as argument
leafNo←leafNo given as argument
S2301
Register cluster indicated by arguments
targetClust←cluster corresponding to clustNo
S2302
Initialize counter i for keyword search
i←0
S2303
Evaluate whether
cluster is leaf-type cluster
targetCluster.type=leaf?
Y
N
S2304
i = key_num?
Y
N
S2305
Search for
gene information:
Is keyword of search[i].keyword
present in the attribute of
gene_info corresponding to
targetClust.geneID?
N
Y
S2306
Add 1 to the number of keyword detected
search[i].times←search[i].times+1
S2307
Register index of detected location
Add leafNo to search[i].place
S2308
Add 1 to search counter of keyword
i←i+1
S2309
Add 1 to index
leafNo←leafNo+1
S2310
Substitute clusterNo of left child
node for clustNo
clustNo←targetClust.left
S2311
Perform Process A:
keyword searching
process on left child node
arguments: clustNo, leafNo
S2312
Evaluate whether
cluster is icon type
targetCluster.type=icon?
Y
N
S2313
Substitute clusterNo of right
child node for clustNo
clustNo←targetClust.right
S2314
Perform Process A:
keyword searching process
on right child node
arguments: clustNo, leafNo
END

Fig. 24

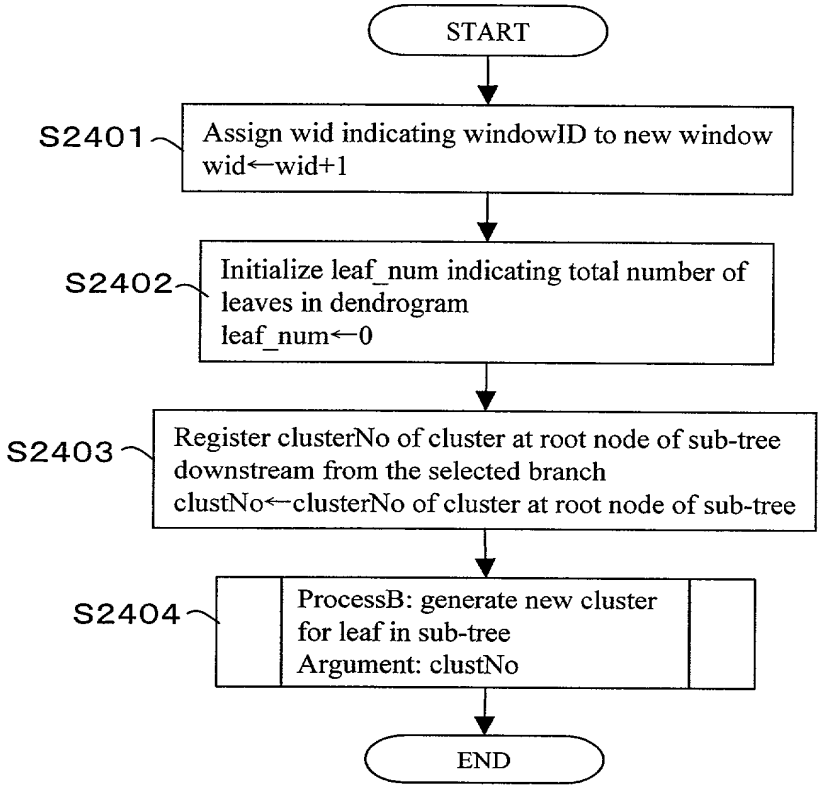
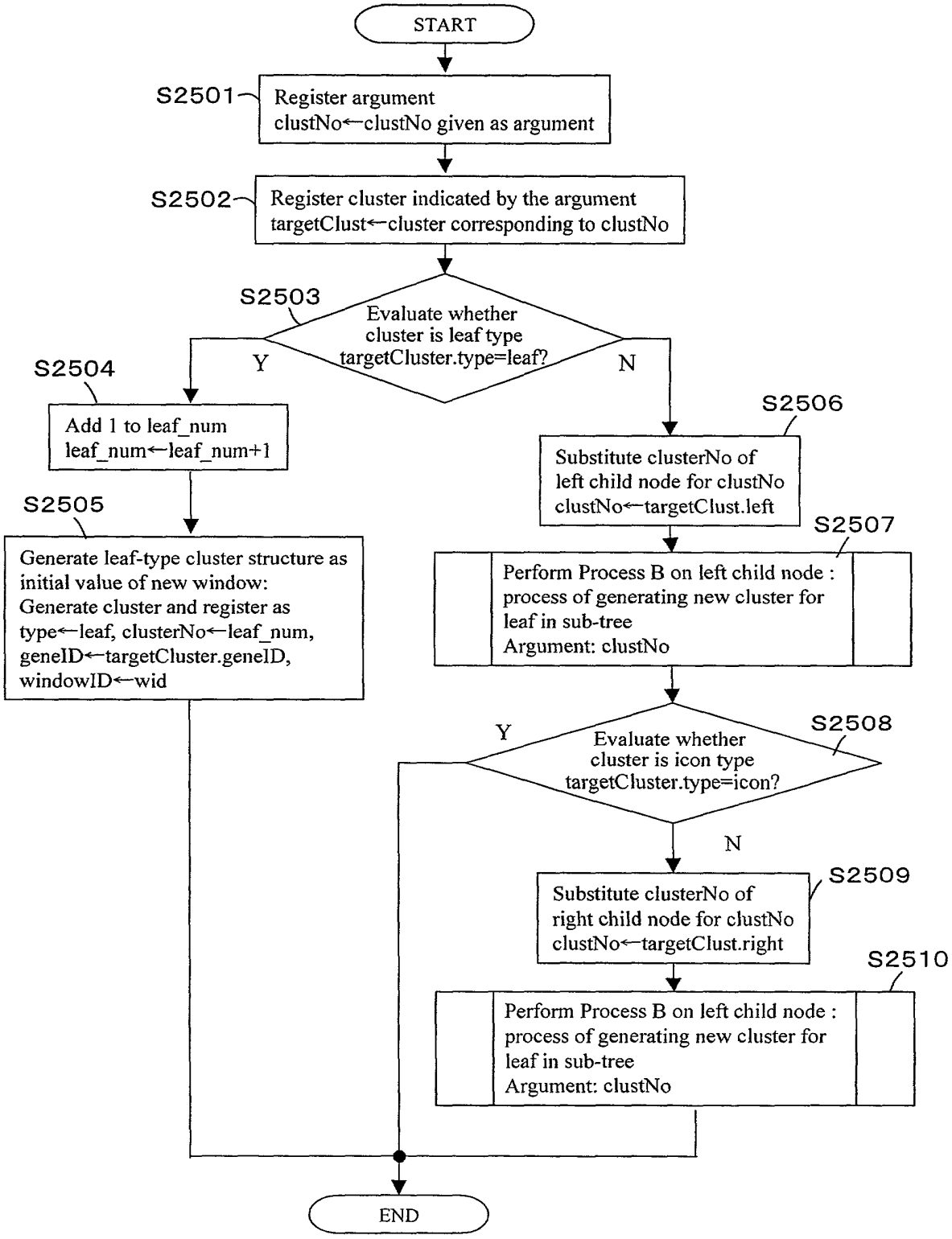

START
S2401
Assign wid indicating windowID to new window
wid←wid+1
S2402
Initialize leaf_num indicating total number of leaves in dendrogram
leaf_num←0
S2403
Register clusterNo of cluster at root node of sub-tree downstream from the selected branch
clustNo←clusterNo of cluster at root node of sub-tree
S2404
ProcessB: generate new cluster for leaf in sub-tree
Argument: clustNo
END

Fig. 25


START
S2501
Register argument
clustNo←clustNo given as argument
S2502
Register cluster indicated by the argument
targetClust←cluster corresponding to clustNo
S2503
Evaluate whether
cluster is leaf type
targetCluster.type=leaf?
Y
N
S2504
Add 1 to leaf_num
leaf_num←leaf_num+1
S2505
Generate leaf-type cluster structure as
initial value of new window:
Generate cluster and register as
type←leaf, clusterNo←leaf_num,
geneID←targetCluster.geneID,
windowID←wid
S2506
Substitute clusterNo of
left child node for clustNo
clustNo←targetClust.left
S2507
Perform Process B on left child node :
process of generating new cluster for
leaf in sub-tree
Argument: clustNo
S2508
Evaluate whether
cluster is icon type
targetCluster.type=icon?
Y
N
S2509
Substitute clusterNo of
right child node for clustNo
clustNo←targetClust.right
S2510
Perform Process B on left child node :
process of generating new cluster for
leaf in sub-tree
Argument: clustNo
END

Attorney's Docket No.:_____

DECLARATION, POWER OF ATTORNEY AND PETITION

I (We), the undersigned inventor(s), hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I (We) believe that I am (we are) the original, first, and joint (sole) inventor(s) of the subject matter which is claimed and for which a patent is sought on the invention entitled

METHOD AND SYSTEM FOR DISPLAYING DENDROGRAM

the specification of which

- ☐ is attached hereto.
- ☐ was filed on _____ as Application Serial No. _____ and amended on _____.
- ■ was filed as PCT international application Number PCT/JP00/08133 on November 17, 2000, and was amended under PCT Article 19 on _____ (if applicable).

I (We) hereby state that I (We) have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above; that I (We) do not know and do not believe that this invention was ever known or used before my invention or discovery thereof, or patented or described in any printed publication in any country before my invention or discovery thereof, or more than one year prior to this application, or in public use or on sale in the United States for more than one year prior to this application; that this invention or discovery has not been patented or made the subject of an inventor's certificate in any country foreign to the United States on an application filed by me or my legal representatives or assigns more than twelve months before this application.

I (We) acknowledge the duty to disclose information known to be material to the patentability of this application as defined in Section 1.56 of Title 37 Code of Federal Regulations.

I (We) hereby claim foreign priority benefits under Section 119(a)-(d) of Title 35 United States Code, of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Application No.	Country	Filing date	Priority claimed
354401/1999	Japan	December 14, 1999	■ Yes ☐ No
			☐ Yes ☐ No
			☐ Yes ☐ No
			☐ Yes ☐ No

I hereby claim the benefit under Section 119(e) of Title 35 United States Code, of any United States application(s) listed below.

(Application Number)	(Filing Date)
(Application Number)	(Filing Date)

I (We) hereby claim the benefit under Section 120 of Title 35 United States Code, of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Section 112 of Title 35 United States Code, I (We) acknowledge the duty to disclose material information as defined in Section 1.56(a) of Title 37 Code of Federal Regulations, which occurred between the filing date of the prior application and national or PCT international filing date of this application:

Application Serial No.	Filing Date	Status (pending, patented, abandoned)

And I (We) hereby appoint:

Paul N. Kokulis, Registration No. 16,773; G. Lloyd Knight, Registration No. 17,698; Kevin E. Joyce, Registration No. 20,508; George M. Sirilla, Registration No. 18,221; Donald J. Bird, Registration No. 25,323; Dale S. Lazar, Registration No. 28,872; Paul E. White, Jr., Registration No. 32,011; Glenn J. Perry, Registration No. 28,458; Kendrew H. Colton, Registration No. 30,368; G. Paul Edgell, Registration No. 24,238; Lynn E. Eccleston, Registration No. 35,861; Timothy J. Klima, Registration No. 34,852; David A. Jakopin, Registration No. 32,995; Mark G. Paulson, Registration No. 30,793; Stephen C. Glazier, Registration No. 31,361; Richard H. Zaitlen, Registration No. 27,248; Roger R. Wise, Registration No. 31,204; Michael R. Dzwonczyk, Registration No. 36,787; W. Patrick Bengtsson, Registration No. 32,456; Jack S. Barufka, Registration No. 37,087; Adam R. Hess, Registration No. 41,835; William P. Atkins, Registration No. 38,821; Paul L. Sharer, Registration No. 36,004; Robin L. Teskin, Registration No. 35,030; Anthony L. Miele, Registration No. 34,393; Robert J. Walters, Registration No. 40,862; Brian J. Beatus, Registration No. 38,825; John R. Wetherell, Registration No. 31,678; Robert M. Bedgood, No. 43,488.

I(We) hereby request that all correspondence regarding this application be sent to the firm of Pillsbury Winthrop LLP whose Post office address is: 11975 El Camino Real, Suite 200, San Diego, CA 92130 U.S.A.

I (We) declare further that all statements made herein of my (our) knowledge are true and that all statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

100 Yasuyuki NOZAKI
NAME OF FIRST SOLE INVENTOR

Yasuyuki Nozaki
Signature of Inventor

July 10, 2001
Date

Residence: Kanagawa, Japan JPK

Citizen of: Japan
Post Office Address:
c/o Hitachi Software Engineering Co., Ltd., 6-81, Onoe-cho, Naka-ku, Yokohama-shi, Kanagawa 231-8475 Japan

200 Tsunehiko WATANABE
NAME OF SECOND JOINT INVENTOR

Tsunehiko Watanabe
Signature of Inventor

July 10, 2001
Date

Residence: Kanagawa, Japan JPK

Citizen of: Japan
Post Office Address:
c/o Hitachi Software Engineering Co., Ltd., 6-81, Onoe-cho, Naka-ku, Yokohama-shi, Kanagawa 231-8475 Japan

300 Ryo NAKASHIGE
NAME OF THIRD JOINT INVENTOR

Ryo Nakashige
Signature of Inventor

July 10, 2001
Date

Residence: Kanagawa, Japan JPK

Citizen of: Japan
Post Office Address:
c/o Hitachi Software Engineering Co., Ltd., 6-81, Onoe-cho, Naka-ku, Yokohama-shi, Kanagawa 231-8475 Japan

4-00

Takuro TAMURA
NAME OF FOURTH JOINT INVENTOR

Signature of Inventor

July 10, 2001
Date

Residence: Kanagawa, Japan JPX

Citizen of: Japan

Post Office Address:
c/o Hitachi Software Engineering Co., Ltd., 6-81, Onoe-cho, Naka-ku, Yokohama-shi, Kanagawa 231-8475 Japan